

LOREM

LOREM

LOREM

LOREM

# LOREM IPSUM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In ac scelerisque nibh. Donec ultricies efficitur ligula, quis laoreet urna suscipit vel. Vivamus congue in massa sed laoreet. Curabitur varius ac enim at pretium.

LOREM IPSUM

## ICSI301 ВЕБ АППЛИКЕЙШН ХИЧЭЭЛИЙН ГАРЫН АВЛАГА





Энэ гарын авлага нь Стэнфордын их сургуулийн [CS142 Web Applications](#) хичээлийн төслийн гарын авлагын орчуулга бөгөөд энэ хичээлийг заадаг, гарын авлагыг бичсэн профессор Мэндел Розенблумаас албан ёсны зөвшөөрлийг авсан болно.

<https://kde-milab.github.io/cs142-website-mn/projects.html> холбоосоор цахим хувилбарыг унших боломжтой.

Орчуулсан: Г.Амарсанаа

Хэвлэлийн хуудас: 4.4

Улаанбаатар, 2025

## Гарчиг

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| Хичээлийн даалгаврыг илгээх .....                                                                     | 1  |
| Урт шүү; уншаад хэрэггүй .....                                                                        | 1  |
| Хэрэггүй файлуудыг хаях/цэвэрлэх .....                                                                | 1  |
| Төслийн хавтсаа шахаж илгээх .....                                                                    | 1  |
| Хавтас, файлын нэр зөв байх .....                                                                     | 2  |
| Төслийн ажлыг хамгаалж дүгнүүлэх .....                                                                | 2  |
| Хиймэл оюуны хэрэгсэл ашиглах ба ёс зүй .....                                                         | 2  |
| MERN программ суулгах заавар .....                                                                    | 4  |
| Node.js суулгах .....                                                                                 | 4  |
| MongoDB суулгах .....                                                                                 | 4  |
| Төсөл 1: HTML ба CSS .....                                                                            | 6  |
| Даалгавар 1 (20 оноо) .....                                                                           | 6  |
| Загварын оноо (5 оноо) .....                                                                          | 8  |
| Нэмэлт оноо (2 оноо) .....                                                                            | 8  |
| Нөөц .....                                                                                            | 9  |
| Зөвлөмж .....                                                                                         | 9  |
| Хүлээлгэн өгөх үр дүн .....                                                                           | 10 |
| Төсөл 2: ЖаваСкрипт дасгал .....                                                                      | 11 |
| Тохиргоо .....                                                                                        | 11 |
| Даалгавар 1: Олон шүүлтүүрт функцийг үүсгэх (10 оноо) ..                                              | 12 |
| Даалгавар 2: Загвар процессор (5 оноо) .....                                                          | 15 |
| Даалгавар 3: cs142-test-project2.js кодыг засаж гадаад<br>орчныг бохирдуулахгүй болгох (5 оноо) ..... | 17 |
| Загварын оноо (5 оноо) .....                                                                          | 17 |

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Зөвлөмж .....                                                              | 17 |
| Илгээх.....                                                                | 18 |
| Төсөл 3: ЖаваСкрипт ба DOM.....                                            | 19 |
| Тохиргоо.....                                                              | 19 |
| Даалгавар 1: ЖаваСкрипт огноо сонгуур (25 оноо).....                       | 19 |
| Даалгавар 2: Хүснэгтийн загвар боловсруулах (15 оноо) ....                 | 21 |
| Нэмэлт шаардлага, зөвлөмж ба бусад.....                                    | 23 |
| Загварын оноо (10 оноо) .....                                              | 25 |
| Илгээх.....                                                                | 25 |
| Төсөл 4: ReactJS хуудас үүсгэх нь .....                                    | 26 |
| Тохиргоо.....                                                              | 26 |
| Эхлэл.....                                                                 | 28 |
| Даалгавар 1: Example харагдацыг ойлгох ба шинэчлэх (5<br>оноо) .....       | 29 |
| Даалгавар 2: Шинэ компонент үүсгэх - муж улсын харагдац<br>(10 оноо) ..... | 30 |
| Даалгавар 3: Байршлыг өөрчлөх (5 оноо).....                                | 31 |
| Даалгавар 4: Харагдацуудыг динамикаар солих (10 оноо) .                    | 31 |
| Даалгавар 5: Нэг хуудаст апп (5 оноо) .....                                | 32 |
| Загвар (5 оноо) .....                                                      | 34 |
| Хүлээлгэн өгөх үр дүн .....                                                | 34 |
| Төсөл 5: Нэг хуудаст апп.....                                              | 35 |
| Тохиргоо.....                                                              | 35 |
| Даалгавар 1: Фото хуваалцдаг апп үүсгэх (40 оноо).....                     | 36 |
| Даалгавар 2: Веб серверээс моделийн өгөгдлийг авах (20<br>оноо) .....      | 40 |
| Загвар (5 оноо) .....                                                      | 42 |

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Нэмэлт оноо (5 оноо).....                                                | 43 |
| Хүлээлгэн өгөх үр дүн .....                                              | 44 |
| Төсөл 6: Аппсервер ба өгөгдлийн сан.....                                 | 45 |
| Тохиргоо.....                                                            | 45 |
| MongoDB өгөгдлийн санг эхлүүлэх .....                                    | 46 |
| Node.js вэб серверийг эхлүүлэх.....                                      | 47 |
| Өгөгдлийн сан ашигладаг веб сервер болгон хөрвүүлэх (40<br>оноо) .....   | 48 |
| Даалгавар 2: Өөрийн программыг axios ашигладаг болгох<br>(10 оноо) ..... | 51 |
| Тест.....                                                                | 52 |
| Нэмэлт оноо (10 оноо) .....                                              | 52 |
| Загварын оноо (5 оноо).....                                              | 53 |
| Хүлээлгэн өгөх үр дүн .....                                              | 54 |
| Төсөл 7: Сесшн ба оролт.....                                             | 55 |
| Тохиргоо.....                                                            | 55 |
| Node.js вэб серверийг эхлүүлэх нь.....                                   | 57 |
| Даалгавар 1: Энгийн нэвтрэлт (15 оноо).....                              | 58 |
| Даалгавар 2: Шинэ сэтгэгдэл (15 оноо).....                               | 60 |
| Даалгавар 3: Зураг байршуулах (15 оноо) .....                            | 61 |
| Даалгавар 4: Бүртгэл ба нууц үг (15 оноо).....                           | 62 |
| Нэмэлт оноо #1: Давсалсан нууц үг (5 оноо) .....                         | 63 |
| Нэмэлт оноо #2: Хөтөчийг шинэчлэх (5 оноо).....                          | 66 |
| Загварын оноо (5 оноо).....                                              | 66 |
| Тест.....                                                                | 67 |
| Зөвлөмж .....                                                            | 67 |
| Хүлээлгэн өгөх үр дүн .....                                              | 72 |

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Төсөл 8: Фото зургийн программ спринт.....                        | 73 |
| Тохиргоо.....                                                     | 73 |
| Даалгавар 1: Зарим түүхийг хэрэгжүүлэх (75 оноо).....             | 73 |
| Зөвлөмж.....                                                      | 75 |
| Дүгнэх.....                                                       | 75 |
| Загварын оноо (5 оноо).....                                       | 77 |
| Илгээхдээ анхаарах зүйлс.....                                     | 77 |
| Загвар (5 оноо).....                                              | 79 |
| Хүлээлгэн өгөх үр дүн.....                                        | 80 |
| Хэрэглэгчийн профайлын дэлгэрэнгүй мэдээллийг өргөтгөх<br>.....   | 81 |
| Сэтгэгдэл дээр @меншион хийх.....                                 | 81 |
| Зургийн харагдах байдлыг хянах.....                               | 82 |
| Үйл ажиллагааны feed.....                                         | 83 |
| Сэтгэгдэл, зураг, хэрэглэгчдийг устгах.....                       | 85 |
| Зурган дээр “like” дарах.....                                     | 85 |
| Хэрэглэгчид таалагдсан зургийн жагсаалт.....                      | 87 |
| Хэрэглэгчдийн шинэ үйл ажиллааг тэмдэглэдэг хажуугийн<br>цэс..... | 89 |
| Зураг таглах.....                                                 | 90 |

# Хичээлийн даалгаврыг ИЛГЭЭХ

## Урт шүү; уншаад хэрэггүй

Хичээлд үүсгэсэн [Тиймс группын](#) Assignments график интерфэйсийг ашиглан төслийн хавтсаар [zip файл](#) үүсгэн илгээнэ. Энэ zip файл дотор нэг ширхэг хавтас байх бөгөөд **projectN** нэртэй байна. Энд **N** бол төслийн дугаар. Жишээ нь, Project1.zip, Project2.zip гэх мэт.

## Хэрэггүй файлуудыг хаях/цэвэрлэх

Ажлаа илгээхээсээ өмнө төслийн хавтаст байгаа шаардлагагүй файлуудыг устгаарай. Төсөлд хэрэглэсэн веб багажууд (tools) хэдэн зуун мегабайт хэмжээтэй файлуудыг төслийн хавтаст чинь үүсгэх бөгөөд эдгээрийг бид авахгүй, илгээх хэрэггүй юм. Дараах хавтаст үүсгэсэн файлуудыг устгасны дараа шахаж zip файл үүсгэнэ.

- **node\_modules** - **package.json** файлд тодорхойлогдсон, npm багажаар татаж авсан модулиуд.
- **compiled** - React.js багажуудын үүсгэсэн ЖаваСкриптүүд

## Төслийн хавтсаа шахаж илгээх

Төслөө хөгжүүлж дуусаад өөрийнхөө өөрчилсөн, нэмсэн бүх файл, дэд хавтсууд чинь бүгд төслийн хавтас дотор орсон эсэхийг шалгаарай. Энэ дотор энгийн горимд дэлгэцлэгдэггүй олон нуугдсан файлууд (hidden files) байдаг шүү. Консол орчноос zip командыг ашиглаад эсвэл хэрэглэгчийн график интерфэйсээр, альнаар нь ч төслийн хавтсаа шахаж zip файл үүсгэж болно.

MacOS системийн хувьд Finder программаас төслийн хавтас дээрээ хулгана/падын баруун товчийг (control-click) дарж гарч ирэх дэд цэснээс "Compress" командыг ашиглаж болно. Виндоус үйлдлийн

системийн хувьд хавтас дээрээ баруун товчийг дарж "Send to..." -> "Compressed (zipped) folder.". Зөвхөн ZIP өргөтгөлтэй файлыг хүлээж авна.

Линакс төрлийн үйлдлийн системийн хувьд терминал программаас, одоо ажиллаж байгаа хавтас дотроо дараах командыг ажиллуулж болно:

```
zip -r projectN.zip projectN
```

Энд **projectN** төслийн нэр болно.

Дээрх арга бүгд **".zip"** өргөтгөлтэй файл үүсгэх бөгөөд үүнийг Тиймсийн assignment-аар илгээгээрэй.

## Хавтас, файлын нэр зөв байх

Төслийн заавар дээр бичсэнээр файлыг яг ижил нэрлэх ёстой. Зарим тохиолдолд ийм нэртэй хавтас дотор тийм нэртэй файл үүсгэх тухай заавар бий. Хэрэв яг ижил нэрээр илгээхгүй бол даалгаврыг шалгаж үнэлэх боломжгүй болно.

## Төслийн ажлыг хамгаалж дүгнүүлэх

Төслийн ажлыг Assignment-аар илгээснээр шалгах үйл явц эхэлнэ. Таны илгээсэн төслийн ажлыг автоматаар шалгана. Ийм учраас хавтсыг шахах, файлыг зөв нэрлэх тухай өмнөх зааврыг хатуу баримтлахыг сануулъя. Хэрэв буруу нэрлэсэнээс болж таны даалгаврыг үнэлээгүй бол та өөрөө хариуцлагаа хүлээх болно шүү. Хэдийгээр автоматаар шалгах аргыг ашиглаж байгаа ч гэсэн багш таныг дуудаж лабораторийн хичээлийн цаг дээр биечлэн хамгаалуулж болно. Энэ тохиолдолд багшийн үнэлгээгээр төслийн ажлыг үнэлэх юм. Төслийн ажлыг хамгаалахдаа өөрийн компьютер дээр ажиллуулж харуулах, асуултад хариулах, кодыг өөрчилж үзүүлэх болон кодыг тайлбарлана. Түүнээс гадна та автоматаар өгсөн үнэлгээтэй санал нийлэхгүй эсвэл биеэр хамгаалж шалгуулахыг хүсвэл багшид хэлээд шалгуулж болно.

## Хиймэл оюуны хэрэгсэл ашиглах ба ёс зүй

Юуны өмнө хэлэхэд, энэ хичээлээр багаар ажиллахгүй, ганцаараа бие даан ажиллаж таслийн ажлыг хийж гүйцэтгэх ёстой. Өөр бусад олон хичээлд багаар ажиллах туршлага, дадлыг олгодог. Харин энэ хичээлээр веб хөгжүүлэх төсөлд багаар ажиллахад өөрийгөө бэлдэж программ бичих үр чадварыг эзэмших юм.

Ёс зүйн хэм хэмжээ

Энэ хичээлээр бид Стэнфордын их сургуулийн Компьютерын ухааны тэнхимийн ёс зүйн дүрмийг баримтална.

"Өөр хүний анхдагч бүтээлийг зохиогч эсвэл эх сурвалжийг үндэслэлтэй, зохих ёсоор дурдахгүйгээр ашиглах, уг бүтээл нь код, томъёо, санаа, үг яриа, судалгаа, стратеги, бичвэр эсвэл бусад хэлбэрийн аль ч байж болно."

- Дүрэм 1: Та өөрийнх бус шийдэл эсвэл программын кодыг харах ёсгүй.
- Дүрэм 2: Та өөрийн шийдлийн кодыг бусад оюутнуудтай хуваалцах ёсгүй.
- Дүрэм 3: Тусламж авсан тохиолдолд, түүнийгээ өөрийн ирүүлсэн ажил дээр заавал тэмдэглэнэ үү.

Бүтээлийн хулгайгаас зайлсхийх зөвлөмж:

- Өөрийн кодыг өөрөө бичих
- Кодыг бусадтай хуваалцах эсвэл хамтран бичихгүй байх
- Ерөнхий стратеги хэлэлцэх, веб технологийн (хэрэгтэй CSS загварууд, программын сангууд, тэдгээрийн функцүүд гэх мэт) талаар хэлэлцэх, мэдлэгээ хуваалцахад бол асуудалгүй.

Хэрэв ChatGPT, Gemini, Claude.ai зэрэг хиймэл оюуны ашиглаад кодоо бичүүлбэл, эсвэл санаа, шийдлийг асууж хэрэглэвэл бүтээлийн хулгай тооцогдоно. Харин үүний оронд хиймэл оюуны хэрэгслээс технологийн талаар асуух, онолын талаар харилцан ярилцах, эсвэл өөрөө кодоо бичиж дууссаны дараа ярилцаж болно.

# MERN программ суулгах заавар

Энэ хичээлээр хийх программын төслүүдэд Node.js болон MongoDB суулгаж ашиглах хэрэгтэй. Хэрэв таны компьютер, нөтбүүк дээр эдгээр программууд суугаагүй бол дараах зааврын дагуу суулгаарай.

## Node.js суулгах

Node.js программын хамгийн сүүлийн "Урт хугацаанд дэмжих - Long Term Support (LTS)" хувилбарыг (одоогоор 18.14.0) суулгана. Үүнийг <https://nodejs.org/en/download> URL хаягаар татаж авах боломжтой. Node.js болон модуль менежер (npm) программыг суусан эсэхийг шалгахдаа доорх командыг ажиллуулна:

```
node -v
```

болон:

```
npm -v
```

энэ нь дээрх хоёр программын хувилбарыг дэлгэцлэж харуулна.

## MongoDB суулгах

[MongoDB Community Edition](#) хувилбарыг <https://docs.mongodb.com/manual/administration/in-stall-community/> хаягаар татаж суулгана.

Дараах командыг ашиглан MongoDB серверийг асаана.

```
mongod (the exact arguments depend on where you placed the database)
```

Мөн доор командаар MongoDB өгөгдлийн санг шууд удирдан ажиллах боломжтой:

```
mongosh
```

Команд промт дээр help гэж бичээд боломжит командуудыг харж болно.

Виндоус системийн хэрэглэгчдийн хувьд MongoDB суусан замыг үйлдлийн системийнхээ environment path хувьсагчид нэмж өгнө. Ихэнхдээ

```
C:\Program Files\MongoDB\Server\<version_number>\bin
```

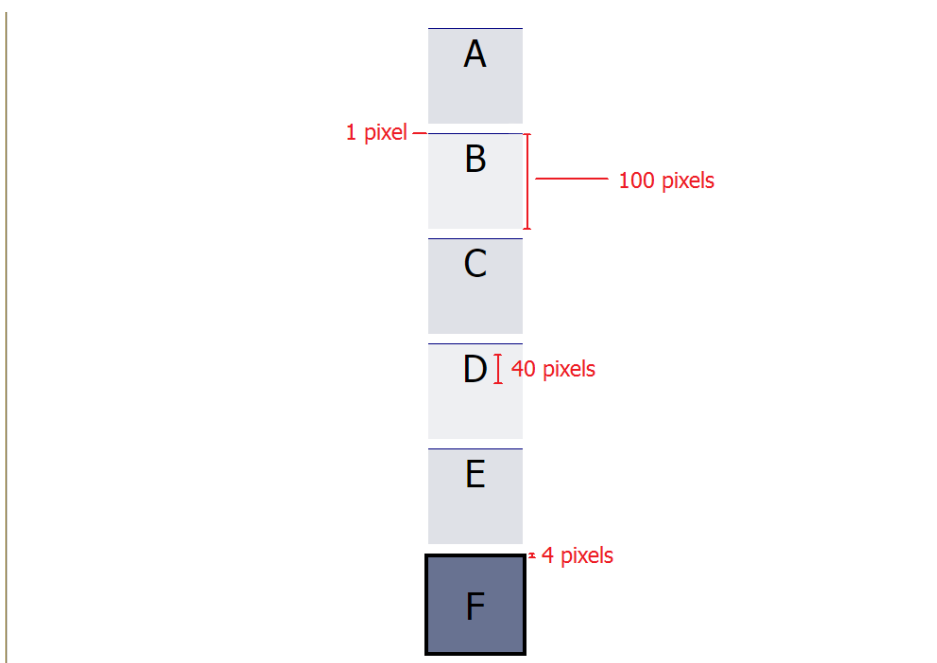
зам дээр суусан байдаг.

# Төсөл 1: HTML ба CSS

## Даалгавар 1 (20 оноо)

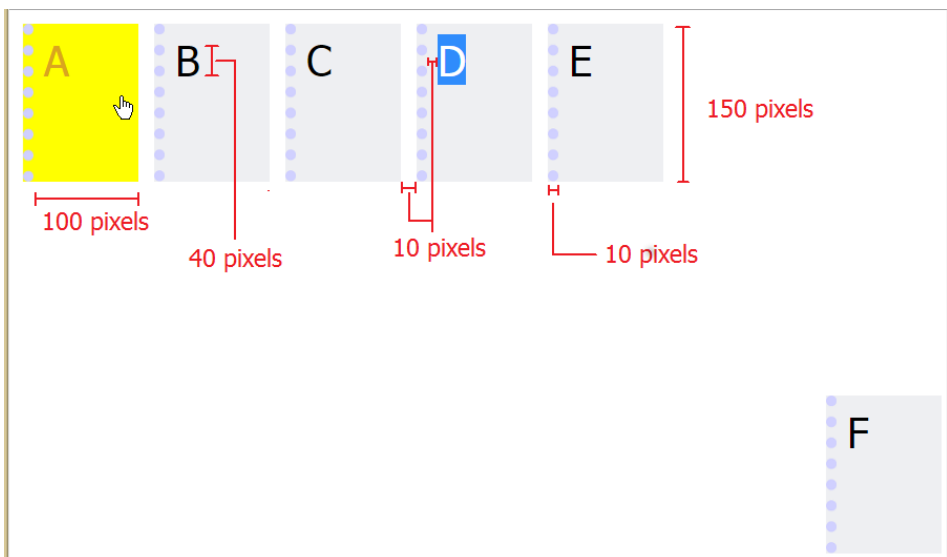
Тодорхойлж өгсөн CSS загварын дагуу хоёр өөр байдлаар харагддаг нэг HTML баримт үүсгэнэ. Энэ HTML файлыг `index.html` гэж нэрлэх ба баримтын гарчиг нь “ICSI301 Төсөл #1” байна. Энэ баримтын зэрэгцээ `styleA.css` болон `styleB.css` гэсэн хоёр загвар (stylesheet) үүсгэнэ.

Хэрэв HTML файл нь `styleA.css` загварыг холбосон бол дараах ("Хувилбар А") зурагт үзүүлсэн шиг харагдана:



Хэрэв HTML файл нь `styleB.css` загварыг холбосон бол дараах ("Хувилбар Б") зурагт үзүүлсэн шиг харагдана:

**Санамж:** Дэлгэцийн агшинд харагдаж байгаа, сонгож идэвхжүүлсэн D үсэг нь бичвэр хэр их зайд гарахыг харуулсан болно. Түүнээс D үсгийг хөх өнгийн дэвсгэртэй загварчлахгүй болно.



### Загвар А тодорхойлолт

- Зургаан хайрцаг элемент босоогоор жагссан байна.
- Бүх хайрцгууд хэвтээ тэнхлэгийн дагуу голлосон, босоо тэнхлэгийн дагуу хоорондоо ижил зайтай байна. Веб хөтчийн цонхны хэмжээг өөрчлөхөд хайрцгуудын хоорондын зай нь өөрчлөгдөнө (хуудасны босоо тэнхлэгийн дагуу ижил хэмжээгээр). Хэдий тийм боловч, хайрцгууд нэг нь нөгөө дэрээ давхарлагдахгүй, хэмжээ нь ч өөрчлөгдөхгүй байх ёстой.
- Хайрцаг бүрийн хэмжээ 100x100 пиксель, дээд талдаа 1px хүрээтэй (өнгө: #687291). Бичвэр нь хэвтээ тэнхлэгийн дагуу голлосон байна.
- Хайрцгууд өнгүүдтэй (#dfe1e7, #eeeff2).
- Сүүлийн элемент (өнгө: #687291) 4px, хар хүрээтэй, доторх бичвэр нь босоо тэнхлэгийн дагуу голлосон байна.
- Бүх элементийн фонт нь Tahoma, 40 пиксель.

### Загвар Б тодорхойлолт

- Таван хайрцаг нь зүүн дээд өнцөгт шахаж байрлах ба хэвтээ тэнхлэгийн дагуу жагссан байна.

- Цонхны хэмжээг өөрчлөхөд хайрцгууд давхарлагдахгүй (жишээ нь. А-гаас Е хайрцгууд нэг шугамдаа байрлана. Цонх нь тэдгээрийг харуулахад хүрэлцэхгүй хэмжээтэй байсан ч давхарлагдахгүй).
- Сүүлийн хайрцаг нь цонхы баруун доод өнцөгт шахаж байрлана. Цонхны хэмжээ өөрчлөхөд тэндээ л байрласан хэвээр байна. Өөрөөр хэлбэл, цонхны хэмжээг дагадаг байна.
- Хайрцаг бүр 100x150 пиксель (өнгө: #eeeff2), зүүн талдаа 10px цэгэн шулуунтай (өнгө: #D0D0FF). Хайрцгууд хоорондоо 10 пиксель зайтай байна.
- Хайрцаг дээр хулгана очиход хулганы заагч нь гарын дүрстэй болно. Мөн дэвсгэр (yellow) болон фонтын (goldenrod) өнгө нь өөрчлөгдөнө.
- Бүх хайрцгийн фонт нь Tahoma, 40 пиксель.
- Хайрцгийн ирмэг болон доторх үсэг нь 10 пиксель зайтай байна.

Хэрэв дээрх тодорхойлолт тусгагдаагүй CSS property байвал, жишээ нь, Загвар Б дэх элементийн хоорондын зай, болон цонхны ирмэг гэх мэт, та өөрийнхөөрөө зохиож хэрэгжүүлж болно.

## Загварын оноо (5 оноо)

Таны HTML файл XHTML 1.0 баримтыг шалгаж баталгаажуулах <http://validator.w3.org> хэрэгслийг ямар нэг алдаа, санамжгүйгээр давсан байх ёстой. Мөн таны HTML, CSS кодууд цэвэрхэн, уншууртай, догол мөрийг зөв авч хэрэглэсэн, зөв бүтэцлэгдсэн байх ёстой.

## Нэмэлт оноо (2 оноо)

Гуравдахь `styleC.css` stylesheet үүсгэж дараахаас хоёрыг нь хэрэгжүүлсэн байна:

- gradients
- transitions
- animations
- web fonts

- shadows

## Нөөц

- Энэ төсөл болоод энэ хичээлээр Chrome веб хөтчийг ашиглана. Энэ нь веб хөтчөөс болж үүсэж болзошгүй аливаа асуудлыг бууруулах юм.
- Төслийнхөө файлуудыг Microsoft Teams программаар, хичээлийн группд даалгавар илгээх хэсгээр илгээнэ.
- Та ямар нэг ЖаваСкрипт эсвэл өөр сан, stylesheet ашиглахгүй. Коддоо зөвхөн өөрөө бичих хэрэгтэй.

## Зөвлөмж

- Энэ төсөлд дараах шиг CSS шинжүүд хэрэгтэй байж болно. (хэрхэн хэрэглэхийг Гүүглдээд гарын авлага, зааврыг нь хараарай):

```
display: inline-block;  
height: 100%;  
white-space: nowrap;
```

- [CSS Flexbox layout](#) модулийг даалгаврын аль нэг хэсэгт ашиглах хэрэгтэй.
- XHTML баримт HTML-ээс яг ямар ялгаатай байна вэ? Жишээ нь, элементийн хэмжээг өгөхөд ялгаатай байна уу.
- position, flex layout-ийн талаар зайлшгүй судлах хэрэгтэй. Эдгээрийг аливаа элементийг өөрийн хүссэнээрээ байршуулахад гол хэрэглэдэг.
- CSS Do Not Repeat Yourself (DRY) зарчмыг сайн баримтлаарай. Илүү оновчтой, дараагийн өөрчлөлтөд уян хатан код бичих хэрэгтэй.
- MDN сайтаас суурь мэдлэгээ заавал аваарай. Ямар нэг элемент, CSS шинжийн талаар судлахдаа үүнийг эхлээд үз. Суурь сайн авбал код бичих цагийг чинь хэмнэх болно.

- Зааврыг бүрэн гүйцэд уншаарай. Анзаараагүй юм байна гээд зарим оюутнууд дутуу хийх тохиолдол байдаг.

## Хүлээлгэн өгөх үр дүн

Даалгаврыг [илгээхдээ](#) Төсөл 1-д шаардлагатай `index.html`, `styleA.css`, `styleB.css`, болон (заавал бус) `styleC.css` файлуудыг оруулаарай.

# Төсөл 2: ЖаваСкрипт

## дасгал

### Тохиргоо

Хэдийгээр энэ төслийг өөрийн веб хөтөч дээрээ ажиллуулж болох ч гэсэн таны үйлдлийн системд Node.js суусан байх хэрэгтэй. Энэ программыг ашиглан кодын чанарыг шалгах юм. Хэрэв суулгаагүй бол Node.js болон npm модуль менежер программыг [зааврын](#) дагуу одоо суулгаарай.

Node.js суулгасны дараа `project2` нэртэй хавтас үүсгээд [энэ zip файлыг](#) дотор нь задлаарай. Энэ zip файлд `cs142-test-project2.html` болон `cs142-test-project2.js` файлууд байх бөгөөд эдгээрийг ашиглаад төслийн кодоо тестлэх фрэймвөрк маягаар ашиглах юм.

Та кодын чанарыг шалгадаг багаж ("linter" гэж нэрлэдэг), [ESLint](#) багажийг татаж аваад `project2` хавтас дотроос дараах командаар ажиллуулна:

```
npm install
```

Энэ нь ESLint багажийг `node_modules` дэд хавтас дотор татаж авна. Та `project2` хавтас дотор байгаа бүх ЖаваСкрипт файлууд үүнийг ажиллуулах боломжтой болох юм. Дараах командаар кодынхоо чанарыг шалгаарай:

```
npm run lint
```

Таны илгээх бүх ЖаваСкрипт файлын код чинь `'use strict';` мөрөөр эхлэх ёстой. Мөн ESLint программаар шалгахад ямар нэг алдаа эсвэл санамж (warning) байх ёсгүй юм шүү. Ямар нэг алдаа эсвэл санамжтай бол загварын онооноос хасагдах болно.

Энэ даалгаварыг ажиллуулахад танд хоёр сонголт байна:

1. **Өөрийн веб хөтөч дээр:** `cs142-test-project2.html` файлыг шууд нээж ажиллуулах. Энэ веб хуудас нь таны бичсэн ЖаваСкрипт файлыг ажиллуулах бөгөөд хэд хэдэн тестийг ажиллуулж үр дүнг харуулна.
2. **Node.js дээр:** Та тестийг веб хөтөчгүйгээр ажиллуулах. Дараах командыг ажиллуулаарай.

```
npm test
```

Бид хөгжүүлэлтийн үед веб хөтчийг ашиглахыг санал болгож байна. Энэ нь программаа алхам бүрээр ажиллуулж үр дүнг нь харахад (debugging) илүү дээр. Гэсэн хэдий ч бид таны төслийг үнэлэхдээ Node.js ашиглана.

**Санамж:** Эдгээр тестүүд нь бүх нөхцлийг тооцоогүй. Эдгээр нь та үндсэн кодоо бичиж дуусмагц программ бүхэлдээ яаж ажилладагийг зааварчилж таниулах болно. Бидний өгсөн файлд байгаа тестүүд даалгаварт өгсөн тодорхойлолтуудыг бүгдийг бүрэн тестлээгүй учир тодорхойлолтын дагуу бүрэн хэрэгжүүлэх эсэх нь таны хариуцах ажил юм шүү.

Энэ төслөөр хэд хэдэн ЖаваСкрипт функц бичнэ. Энэ бие даалтаар өгч буй даалгаваруудаар хөгжүүлэх бодит практик болон функцүүд нь хичээлийн бусад төслүүдэд дараа хэрэг болно. Өгсөн ЖаваСкриптийн ямар даалгаврыг гүйцэтгэхэд шаардлагатай ЖаваСкриптийн зарим стандарт функцүүдийг хэдэн мөрөнд ашиглаж болно. Энэ төслийн зорилго нь ЖаваСкриптийг сурахад чиглэж байгаа тул ЖаваСкриптийн бусад бэлэн санг **ашиглахыг зөвшөөрөхгүй**. Зөвхөн ЖаваСкриптийн өөрийнх нь функцүүдийг (built-in), жишээ нь, Arrays, Date Objects гэх мэт, ашиглахыг зөвшөөрнө.

## Даалгавар 1: Олон шүүлтүүрт функцийг үүсгэх (10 оноо)

`project2` хавтас дотроо `cs142-make-multi-filter.js` нэртэй шинэ файл үүсгэнэ. Таны үүсгэх Олон шүүлтүүрт функцийн код энэ файлд байна.

`cs142MakeMultiFilter` нэртэй гадаад функц тодорхойлно. Энэ нь нэг хүснэгтэн хувьсагч (`originalArray`) параметртээ авч тухайн хүснэгтэн хувьсагчийн элементүүдийг шүүж гаргахад хэрэглэх нэг функц буцаадаг байна. Буцааж буй (`arrayFilterer`) функц нь дотроо `currentArray` гэж нэрлэх хувьсагчид шүүгдэж үлдсэн элементүүдийг хадгалж явдаг байх юм. Эхэндээ `currentArray` нь `originalArray` хувьсагчтай ижил утгатай байна. `arrayFilterer` функц нь хоёр функцийг параметртээ авна. Үүнд:

1. `filterCriteria` - Энэ функц параметртээ нэг хүснэгтэн хувьсагчийг авч `boolean` утга буцаадаг. Энэ функц `currentArray` хувьсагчийн элемент бүрийн хувьд дуудагдах бөгөөд `currentArray` нь `filterCriteria` функцийн үр дүнд шинэчлэгддэг байна. Хэрэв `filterCriteria` функц тухайн элементийн хувьд `false` утга буцаавал тэр элемент `currentArray` хувьсагчаас хасагдана. Эсрэг тохиолдолд тухайн элемент нь `currentArray` хувьсагчдаа үлдэнэ. Хэрэв `filterCriteria` функц биш байвал буцааж буй (`arrayFilterer`) функц `currentArray` хувьсагчийн утгыг ямар нэг шүүлт хийлгүйгээр шууд буцаадаг байна.
2. `callback`- Энэ функц нь шүүлт хийгдэж дуусмагц дуудагддаг. `callback` нь `currentArray` хувьсагчийн утгыг аргументдээ авдаг. `callback` функц дотроос `this` хувьсагчаар хандахад `originalArray` хувьсагчийн утгыг заадаг байх ёстой. Хэрэв `callback` функц биш бол үүнийг дуудахгүй. `callback` нь ямар нэг утга буцаахгүй.

`arrayFilterer` функц нь `filterCriteria` параметр тодорхойлогдоогүй, `currentArray` хувьсагчийг буцаагаагүй бусад нөхцөлд өөрийгөө буцаадаг байна. Энэ нь `arrayFilterer` функцийг нэг агшинд олон удаа ажиллуулах боломжийг олгодог байна.

Дараах кодоор таны тодорхойлсон функцийг хэрхэн ашиглаж болохыг харууллаа:

```
// cs142MakeMultiFilter() функцийг originalArray = [1, 2, 3]
// параметртэйгээ дуудахад нэг функц буцаана,

// энэ нь arrayFilterer1 хувьсагчид хадгалагдана, үүнийг олон
// давтамжтай дуудаж
```

```

// оролтын хүснэгтэн хувьсагчийн элементийг шүүнэ.
var arrayFilterer1 = cs142MakeMultiFilter([1, 2, 3]);

// arrayFilterer1 функцийг (callback функц өгч) дуудаад 2-той
// тэнцүү биш утгуудыг шүүнэ.
arrayFilterer1(function (elem) {
    return elem !== 2; // элемент 2-той тэнцүү биш эсэхийг шалгах
}, function (currentArray) {
    // callback функц доторх 'this' хувьсагч [1, 2, 3] утгатай
    // originalArray-г заана.
    console.log(this); // [1, 2, 3] гэж дэлгэцлэнэ
    console.log(currentArray); // [1, 3] гэж дэлгэцлэнэ
});

// arrayFilterer1 функцийг (callback функцгүй) дуудахад 3-тай
// тэнцүү биш утгуудыг шүүнэ.
arrayFilterer1(function (elem) {
    return elem !== 3; // элемент 3-тай тэнцүү биш эсэхийг шалгах
});

// arrayFilterer1 функцийг filterCriteria параметргүй дуудахад
// currentArray-г буцаана.
var currentArray = arrayFilterer1();
console.log('currentArray', currentArray); // [1] гэж дэлгэцлэнэ
. Учир нь 2 болон 3 утгууд хасагдсан.

// arrayFilterer өөрийгөө буцаадаг тул, ар араас нь гинжилж дууд
аж болно.
function filterTwos(elem) { return elem !== 2; }
function filterThrees(elem) { return elem !== 3; }

```

```

var arrayFilterer2 = cs142MakeMultiFilter([1, 2, 3]);
var currentArray2 = arrayFilterer2(filterTwos)(filterThrees)();
console.log('currentArray2', currentArray2);
// [1] гэж дэлгэцлэнэ. Учир нь 2 болон 3 утгууд хасагдсан.

// Олон шүүлтүүрт функцийг нэг агшинд олон удаа нь хэрэглэнэ.
var arrayFilterer3 = cs142MakeMultiFilter([1, 2, 3]);
var arrayFilterer4 = cs142MakeMultiFilter([4, 5, 6]);
console.log(arrayFilterer3(filterTwos)()); // [1, 3] дэлгэцлэнэ
console.log(arrayFilterer4(filterThrees)()); // [4, 5, 6] дэлгэцлэнэ

```

## Даалгавар 2: Загвар процессор (5 оноо)

`project2` хавтас дотроо `cs142-template-processor.js` нэрээр нэг файл үүсгэнэ. Загвар процессорын код энэ файлд байх ёстой.

Нэг Загвар процессор класс үүсгэнэ (`cs142TemplateProcessor`). Энэ нь нэг `string` (тэмдэгт мөр) төрлийн `template` параметр авдаг байгуулагчтай, `fillIn` нэртэй нэг методтой байна. Энэ `fillIn` методыг `dictionary` объект өгч дуудахад тэмдэгт мөр буцаах бөгөөд энэ утгыг `dictionary` объектын утгуудаар загварыг (`template`) дүүргэсэн байна. `cs142TemplateProcessor` классыг ЖаваСкриптын стандарт байгуулагч болон `prototype` бүтцийг ашиглан бичнэ.

`fillIn` метод `template` тэмдэгт мөрийг буцаах ба `{{property}}` (шинж) хэлбэртэй/форматтай ямар ч бичвэрийг методод дамжуулж өгсөн `dictionary` объектын харгалзах `property` утгуудаар сольдог байна.

Хэрэв загвар нь `dictionary` объектод байхгүй шинжийг агуулж байвал тэр шинж байх газар хоосон тэмдэгтээр солино. Хэрэв шинж нь хоёр үгийн хооронд байрласан бол хоёр хоосон зай харагдах болно. Жишээ нь: `"This {{undefinedProperty}} is cool"` → `"This is cool"`. Энэ бол байж болно. Та нэмэлт хоосон зайг яах талаар санаа зовох хэрэггүй.

Таны систем зөвхөн зөв форматтай шинжийг л авч үздэг байна. Магадгүй энэ нь дараах тохиолдолд, бид нэг бүрчлэн шалгаагүй учир undefined байж болзошгүй.

- давхарлагдсан шинжүүд - `{{foo {{bar}}}}` эсвэл `{{{{bar}}}}` эсвэл `{{{bar}}}`
- ижил биш уян хаалт - `{{bar}}`
- шинжийн нэрэнд байх уян хаалт - `da{y or da}y`

Дараах кодоор энэ даалгавраар тодорхойлж буй функцүүдийг хэрхэн ашиглаж болохыг харууллаа:

```
var template = 'My favorite month is {{month}} but not the day {
{day}} or the year {{year}}';
var dateTemplate = new cs142TemplateProcessor(template);

var dictionary = {month: 'July', day: '1', year: '2016'};
var str = dateTemplate.fillIn(dictionary);

assert(str === 'My favorite month is July but not the day 1 or t
he year 2016');
```

//тохиолдол: шинж dictionary объектод байхгүй байх

```
var dictionary2 = {day: '1', year: '2016'};
var str = dateTemplate.fillIn(dictionary2);

assert(str === 'My favorite month is  but not the day 1 or the y
ear 2016');
```

## Даалгавар 3: `cs142-test-project2.js` кодыг засаж гадаад орчныг бохирдуулахгүй болгох (5 оноо)

Бидний өгсөн тестийн ЖаваСкрипт файл (`cs142-test-project2.js`) нь нэлээд олон хувьсагчийг ЖаваСкриптийн гадаад орчинд зарласан байгаа. Жишээ нь, энэ файлыг дуудахад `p1Message` хувьсагч гадаад орчинд харагдана. Иймд өөр нэг ЖаваСкрипт файлаас `p1Message` энэ хувьсагчид хандах боломжтой гэсэн үг. `cs142-test-project2.js` файлын кодыг ЖаваСкриптийн стандарт модулийн **anonymous function** ашигладаг загвараар гадаад орчинд байх хувьсагчуудыг нуудаг, бас таны бичсэн кодууд өмнөх шиг шалгаж болдог байхаар өөрчилнө.

## Загварын оноо (5 оноо)

Энэ оноог таны бичсэн ЖаваСкрипт код цэвэрхэн, уншууртай, ESLint санамжгүй тохиолдолд өгнө.

## Зөвлөмж

- ЖаваСкрипт хэлэнд тэнцүү/гэнцүү бишийг шалгахад `==` болон `!=` операторуудын оронд `===` болон `!==` операторыг ашиглаарай.
- ESLint программыг ажиллуулах үед `.eslintrc.json` (нуугдмал файл) байгаа эсэхийг шалгаарай. Үгүй бол ESLint алдаа өгнө. `ls -a (dir /a` Виндоус дээр) командын мөрөөр тухайн хавтас дотор байгаа бүх файлыг, нуугдмал файлыг ч харж болно.
- Хэрэв ESLint олон тооны загварыг асуудлуудыг гаргавал `npm run lint -- --fix` командаар ихэнхийг нь засах боломжтой.
- Файлуудаа хуулахдаа командын мөрийг ашиглаарай. Чирээд хуулах нь ихэвчлэн нуугдмал файлыг хуулдаггүй нь `zip` файлд орох ёстой `.eslintrc.json` файлгүйгээр илгээх нөхцлийг үүсгэдэг.

- Даалгавар 1-д үзүүлсэн шаардлага `cs142MakeMultiFilter` функцийг нэгэн зэрэг олон удаа ашиглах тул `currentArray` хувьсагчийг гадаад орчинд хадгалахгүй байх ёстой.
- Даалгавар 3-т, зөвхөн тест файлд байгаа хувьсагчуудтай л ажилла. Бас `cs142Project2Results` хувьсагчийг гадаад орчинд нь (жишээ нь, `window` объектын шинжээр) үлдээх ёстой. Ийм байж `run-tests-using-node.js` скриптыг ажиллуулж болно.
- Даалгавар 1-ийн хувьд:
  - ЖаваСкрипт функц, closure, callback функц, function prototype, this хувьсагч зэргийг онцгойлон судлах
  - Бичих функцыг чинь яаж ашиглаж байгаа кодыг сайтар ойлгох, `console.log` ашиглан debug хийж үзэх
- Даалгавар 2-ийн хувьд: объект, класс, класс тодорхойлох функцийн арга, прототайп, string, regular expression зэргийг зайлшгүй судлах бичвэр дотор өгөгдөх паттерн, жишээ нь, `{{month}}` нь дурын паттерн байж болохоор хийх
- Даалгавар 3-ийн хувьд:
  - global болон local хувьсагчийн үйчлэх хүрээ, anonymous function, immediately invoked function expression (IIFE) бичиглэл зэргийг тусгайлан үзэх
  - ЖаваСкрипт программын модуль хэрхэн бичдэгийг ойлгох
- Нийтлэг зөвлөмж:
  - Зөвхөн өөрөө бодож, шийдэл боловсруулж, хичээл зүтгэлээр хийх
  - Интернэтээс эсвэл бусдаас бэлэн шийдэл, энэ даалгаварыг хийсэн код харахгүй байх (ингэвэл өөрөө сэтгэхээ больдог шүү)
  - Өөрийн хэрэгжүүлсэн код бүрийг тайлбарладаг байх нь чухал

## Илгээх

Даалгаврыг `илгээхдээ` `project2` нэртэй хавтас дотор `cs142-make-multi-filter.js`, `cs142-template-processor.js`, `cs142-test-project2.js`, болон `cs142-test-project2.html` файлуудыг оруулсан байна. Эдгээр файлуудад `npm run lint` болон `npm test` командуудыг ажиллуулсан байх ба ямар нэг алдаа, санамжгүй болсон байх ёстой.

# Төсөл 3: ЖаваСкрипт ба DOM

## Тохиргоо

Хэдийгээр энэ төслийг өөрийн вэб хөтөч дээрээ ажиллуулж болох ч гэсэн таны үйлдлийн системд Node.js суусан байх хэрэгтэй. Энэ программыг ашиглан кодын чанарыг шалгах юм. Хэрэв суулгаагүй бол Node.js болон npm модуль менежер программыг [зааврын](#) дагуу одоо суулгаарай.

Node.js суулгасны дараа `project3` нэртэй хавтас үүсгээд [энэ zip файлыг](#) дотор нь задлаарай.

Та кодын чанарыг шалгадаг багаж ("linter" гэж нэрлэдэг), [ESLint](#) багажийг татаж аваад `project3` хавтас дотроос дараах командаар ажиллуулна:

```
npm install
```

Энэ нь ESLint багажийг `node_modules` дэд хавтас дотор татаж авна. Таны `project3` хавтас дотор байгаа бүх ЖаваСкрипт файлууд дээр үүнийг ажиллуулах боломжтой болох юм. Дараах командаар кодынхоо чанарыг шалгаарай:

```
npm run lint
```

Таны илгээх бүх ЖаваСкрипт файлын код чинь `'use strict';` мөрөөр эхлэх ёстой. Мөн ESLint программаар шалгахад ямар нэг алдаа эсвэл санамж (warning) байх ёсгүй юм шүү. **Ямар нэг алдаа эсвэл санамжтай бол загварын онооноос хасагдах болно.**

## Даалгавар 1: ЖаваСкрипт огноо сонгуур (25 оноо)

Энэ даалгаврын хувьд та HTML, CSS, ЖаваСкрипт гурвыг хослуулан хоёр `DatePicker` [ЦОНХЫГ](#) хэрэгжүүлэх юм.

`project3` хавтас дотор `DatePicker` нэртэй JavaScript классыг хэрэгжүүлэх `DatePicker.js` файлыг үүсгэнэ үү. Тухайн классыг хэрхэн ашиглаж болохыг дараах жишээн дээрээс харна уу:

```
var datePicker = new DatePicker("div1", function (id, fixedDate)
{
    console.log("DatePicker with id", id,
        "selected date:", fixedDate.month + "/" + fixedDate.day +
        "/" + fixedDate.year);
});
datePicker.render(new Date("July 4, 1776"));
```

Уг классын байгуулагч функц нь өмнө үүссэн `div` элементийн `id` атрибут болон огноо сонгодог `callback` функцийг аргументээрээ авна. Хэрэглэгч огноог сонгох үед `callback` функц нь `id` болон `fixedDate` объектийг хувьсагчаараа авч дуудагдах ёстой. `fixedDate` объект нь дотроо `month`, `day`, `year` гэх шинжүүдтэй бөгөөд харгалзах он сарыг тоогоор илэрхийлсэн байна. (жишээ нь: `{month: 1, day: 30, year: 2016}` объект нь 2016 оны 1 сарын 30-ны өдрийг илэрхийлнэ.)

Энэ классаас үүсэх объект нь тодорхой сарыг заах `Date` объектыг аргументээрээ авдаг `render` функцтэй байх ёстой (`Date` объект нь тухайн сарын доторх дурын өдрийг илэрхийлж болно). `render` функцийг ажиллуулах үед энэ нь огноо сонгогчийн `div` элементийн доторх агуулгыг аялал жуулчлалын захиалгын вэбсайт зэрэгт түгээмэл ашигладаг нэг сарын хуанлийн HTML баримтаар солино.

- Хуанли нь сарын өдрүүдийг долоо хоног бүрээр нэг мөр, гараг бүрээр нэг багана болгон хүснэгт хэлбэрт харуулна.
- Гарагийн багана ням гарагаас эхэлж бямба гарагт дуусдаг байна. Хуанли нь толгой хэсэгтээ харгалзах гарагийн товчилсон нэрийг харуулдаг мөртөй байх хэрэгтэй. Жишээлвэл "Ня", "Да", "Мя" гэх мэт.

- Тухайн сарын өдөр бүрийг тоогоор илэрхийлнэ.
- Зарим долоо хоногийн мөрөнд тухайн сард багтаагүй өдөр орох тохиолдол гарна. Эдгээр өдрүүдийг өөр сарын өдөр гэдгийг нь илтгэн өнгийг нь бууруулж ялгаж өгнө.
- Долоо хоногийн мөрүүд нь сонгогдсон сарын ядаж нэг өдрийг агуулдаг байна. Ихэнхи саруудад долоо хоногийн мөр 5 ширхэг байх боловч, зарим саруудын өдрөөс хамаарч 4 эсвэл 6 мөртэй байх тохиолдол гарна. Таны хийсэн хуанлийн мөрүүдийн тоо нь хувьсдаг байх ёстой.
- Хуанли нь дээд хэсэгтээ сонгогдсон жил болон сарыг харуулдаг байна. Мөн дээрээс нь, өмнөх эсвэл дараах сар руу шилжих боломжтой "<" болон ">" товчнуудтай байна.
- Тухайн сарын өдөр дээр дарах үед дээр дурьдсан callback функцийг тохирох аргументээр дуудаж ажиллуулдаг байна. Сонгогдсон сард хамаарахгүй буюу өмнөх болон дараах сарын өдөр дээр дарах үед callback функц дуудагдах ёсгүй.

Бидний өгсөн `datepicker.html` файл нь дотроо хоёр хоосон `div` элементтэй, мөн уг элементүүдэд хуанлиг харуулдаг `DatePicker` классыг хэрэгжүүлэх багахан ЖаваСкрипт кодыг агуулна. Нэг хуанли нь одооны он сарыг харуулдаг бол нөгөөх нь 2009 оны 1 сарыг харуулдаг байна. Хоёр хуанлийн сарыг сонгох товчлуурууд нь хоорондоо *хамааралгүйгээр*, тусдаа ажилладаг байх хэрэгтэй.

Өгөгдсөн HTML файл нь ямар нэгэн загваргүй байгаа учир `datepicker.css` нэртэй stylesheet үүсгэн, хуанлинуудыг хэрэглэхэд эвтэйхэн болгон загварчилна уу. Stylesheet болон HTML файл хоёрыг холбох `link` тагийг аль хэдийн өгөгдсөн файлд бичиж өгсөн байгаа.

## Даалгавар 2: Хүснэгтийн загвар боловсруулах (15 оноо)

`TableTemplate.js` файлыг үүсгэж, `TableTemplate` нэртэй, дотроо `fillIn` гэсэн **статик** методыг агуулдаг ЖаваСкрипт классыг хэрэгжүүлнэ үү.

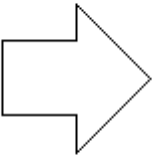
`fillIn` методод тавигдах шаардлагууд:

- Энэхүү метод нь `<table>` элементийн `id` атрибут, `dictionary` объект, мөн `columnName` нэртэй тэмдэгт мөрөөс бүрдсэн гурван аргумент авдаг байна.
- Энэ метод нь юун түрүүнд эхний мөрөн дэхь `{{property}}` ийм хэлбэрт байгаа бичвэрийг `dictionary` объектын харгалзах `property` (шинж чанараар) сольж бичих юм. Үүний дараа `columnName` аргументтэй таарж буй баганы доторх `{{property}}` элементүүдийг өөрчилж бичнэ.
- Уг метод нь хүснэгтийн эхний мөрөнд баганы нэрнүүд байгаа гэж үзэн ажилладаг байна. Оролтоор орж ирсэн хүснэгтийн баганы нэрнүүд хэзээ ч давхцахгүй гэж тооцно уу.
- Хэрэв `columnName` аргументыг тодорхойлж өгөөгүй бол уг метод нь хүснэгтийг бүхэлд нь боловсруулдаг байна.
- Хэрэв дамжуулж өгсөн `columnName` нь таараагүй тохиолдолд уг метод нь тухайн багана доторх бичвэрүүдийг өөрчлөх ёсгүй. Гэхдээ `columnName` нь таараагүй байсан ч эхний мөр дэхь загвар тэмдэгтүүдийн утгыг өөрчлөх хэрэгтэй гэдгийг санаарай.

Жишээлбэл: `columnName` аргументтэй таарж буй баганад байрлах `td` элементийн утга нь "Өнөөдөр бол `{{year}}` оны `{{month}}`-р сарын `{{day}}`-ны өдөр." гэж байгаад, `dictionary` объект аргументын утга нь `{year: "2016", month: "1", day: "30"}` хэмээн өгөгдөхөд `td` элементийн утга нь "Өнөөдөр бол 2016 оны 1-р сарын 30-ны өдөр" болж өөрчлөгдөнө.

Зүүн талын хүснэгтийн `id` атрибут нь `"table"` бөгөөд, `dict` нь тэмдэгтэн мөрөөс бүрдсэн `dictionary` объект байх үед `TableTemplate.fillIn("table", dict, 'Part Number')` уг методыг харгалзах аргументээр дуудсаны дараа баруун талын хүснэгт үүсэх ёстой:

| {{PartNumber}} | {{Length}} |
|----------------|------------|
| {{n14926}}     | {{n47}}    |
| {{n773}}       | {{n3_5}}   |
| {{n9318}}      | {{n10}}    |
| {{n3045}}      | {{n4}}     |



| Part Number | Length   |
|-------------|----------|
| 14926       | {{n47}}  |
| 773         | {{n3_5}} |
| 9318        | {{n10}}  |
| 3045        | {{n4}}   |

original table                      filled-in table

Хэрэв хүснэгтийн загвар дотор `dictionary` объектод тодорхойлогдоогүй `property` байвал түүний оронд хоосон тэмдэгт оруулж өгнө. Таны хэрэгжүүлсэн метод нь зөвхөн зөв хэлбэрт оруулсан загваруудтай ажилладаг байхад хангалттай. Хэд хэдэн загварууд давхардах (ж.нь: `{{foo {{bar}}}}`) эсвэл `{{` хаалт орхигдох г.м) алдаатай загвар өгөгдөхөд боловсруулах шаардлаггүй. Давхардсан хүснэгтүүд болон олон хэсгээс бүрдэх төвөгтэй хүснэгтийн нүд зэрэгтэй мөн ажиллах шаардлаггүй.

Төсөл 2-н хүрээнд бичсэн `ICSI301-template-processor.js` кодоо дахин ашиглан `fillIn` методыг хэрэгжүүлээрэй. `project3` хавтас доторх HTML файлыг нээж үзэж, `script` тагийг хэрхэн ашиглан `ICSI301-template-processor.js` кодыг оруулж ирж буйг ажиглана уу.

Хэрэв хүснэгтэнд тань `<tbody>` элемент байхгүй бол вэб хөтчүүд автоматаар `<tr>` элементүүдийн гадуур нь `<tbody>` нэмж өгдөгийг анхаараарай.

Таны хэрэгжүүлсэн метод хүснэгтийг бүхэлд нь боловсруулж дууссаны дараа хүснэгтийн загварын `visibility` шинж чанарыг шалгаад хэрэв утга нь `hidden` байвал `visible` болгон өөрчилнө үү.

Дээрх ЖаваСкрипт классыг үүсгэсний дараа `ICSI301-test-table.html` файлыг вэб хөтөч дээрээ нээнэ үү. Уг HTML файл нь дотроо жишиг хүснэгтүүд зэргийг агуулдаг бөгөөд таны бичсэн кодыг ажиллуулж, үр дүнг харуулдаг юм. Энэхүү файлыг өөрчилж болохгүй.

## Нэмэлт шаардлага, зөвлөмж ба бусад

- Уг даалгаврын хүрээнд тодорхойлох ЖаваСкрипт классуудыг ECMAScript 2015 **класс** [бичиглэлийг](#) ашиглан, эсвэл Төсөл 2-т хэрэглэсэн класс тодорхойлох аргуудаар бичиж болно.
- Кодоо бичихдээ jQuery болон бусад бүх ЖаваСкрипт бэлэн сангуудыг ашиглаж болохгүй. Жинхэнэ ажил дээр вэб хөгжүүлэх үедээ мэдээж сангууд ашиглаж болох ч, энэ даалгавраар танд ЖаваСкрипт/ДОМ хэрхэн ажилладаг талаар суурь мэдлэгийг олгох нь гол зорилго юм.
- Харин ЖаваСкрипт доторх төрөлх функцүүдийг ашиглаж болно. Жишээлвэл дараах функцүүд хэрэг болно:
  - **Date** классын методууд
  - **parseInt**, **parseFloat**, **isNaN**.
- Дараах ДОМ элементийн шинж чанарууд хэрэг болж болох юм:
  - **cells**
  - **firstChild**
  - **nextSibling**
  - **rows**
  - **tagName**
  - **textContent**
  - **innerHTML**
- Бас нэгэн debug хийх үед ихээхэн хэрэгтэй ЖаваСкрипт функц бол **console.log** юм. Уг функц нь тэмдэгтэн мөрийг аргументээр авч, ЖаваСкрипт log хэсэгт хэвлэн харуулдаг. Энэхүү log буюу бүртгэлийг Chrome хөтчийн control menu дотроос "More Tools → Developer Tools → JavaScript console" харж болно. Бичсэн ЖаваСкрипт код тань ойлгомжгүй байвал **console.log** тусламжтай кодын аль хэсэг дуудагдаж байгааг төвөггүй мэдэх боломжтой.
- Мөн debug хийж байх үедээ **debugger** түлхүүр үгийг ашиглаж болох юм. Хэрэв кодонд "**debugger;**" мөр дуудагдвал программыг ЖаваСкрипт debugger-рүү шилжүүлдэг. Энэ нь кодын замаар хэрэгжүүлж болох breakpoint гэсэн үг юм.
- Хэрэв өөрчлөлт ажиглагдахгүй байвал ЖаваСкрипт кодонд тань алдаа байгаа бөгөөд түүнээс болоод код тань дуудагдахгүй байж болох юм. Тийм эсэхийг шалгахын тулд ЖаваСкрипт console цонхыг нээж error мэдэгдэл байгаа эсэхийг шалгана уу.

- ЖаваСкрипт console болон Chrome DevTools хэрэглүүрүүд нь дараа дараагийн даалгавруудад таны хамгийн дотны найзууд байх болно. Энэ даалгаврын хүрээнд тодорхой цаг зарцуулан эдгээр багажуудыг хэрэглэж дадхыг санал болгож байна.
- ESLint программыг ажиллуулах үед `.eslintrc.json` (далд файл) байгаа эсэхийг шалгаарай. Эс бөгөөс ESLint алдаа заах болно.
- Хэрэв ESLint ажиллаад, олон алдаа илрүүлвэл `npm run lint -- --fix` командаар дийлэнхийг нь засах боломжтой.

## Загварын оноо (10 оноо)

ЖаваСкрипт код тань цэвэрхэн (классуудыг зохистой хэрэглэсэн, байгуулагч функцаас бусад глобал хувьсагч байхгүй, г.м), хялбар уншигдахуйц, мөн ESLint алдаагүй байх хэрэгтэй.

## Илгээх

`project3` хавтсыг бүхэлд нь [илгээнэ](#).

# Төсөл 4: ReactJS хуудас

## ҮҮСГЭХ НЬ

### Тохиргоо

Та өөрийн систем дээрээ Node.js болон npm модуль менежерийг суулгасан байх ёстой. Хэрэв үгүй бол, [суулгах](#) зааврыг дагаж одоо суулгаарай.

`project4` нэрээр хавтас үүсгэж [энэ файлыг](#) дотор нь задалж хуулаарай. Энэ `zip` файл энэ даалгаварт ашиглах гарааны файлуудыг агуулж байгаа.

Энэ даалгаварт нэлээдгүй `node` модулиуд (жишээ нь, [Webpack](#), [Babel](#), [ESLint](#)) шаардлагатай. Эдгээр багаж [ReactJS](#) веб апп босгох (build), мөн веб аппыг веб хөтчид өгч үйчлэх энгийн Node.js веб сервер ([ExpressJS](#)) үүсгэхэд хэрэг болно. Эдгээр модулийг `project4` хавтас дотроос дараах командыг ажиллуулж суулгана:

```
npm install
```

Энэ команд ойролцоогоор 100 мегабайт хэмжээтэй 600 `node` модуль `node_modules` гэсэн дэд хавтас дотор татаж авна.

Бид `npm` командыг янз бүрийн багажууд татаж авахад хэрэглэдэг. `package.json` файлын `"scripts"` шинжид харагдаж байгаа шиг дараах командуудыг ажиллуулах боломжтой:

- `npm run build` - [Webpack](#) багажийг `webpack.config.js` файлд байгаа тохиргоогоор ажиллуулж төсөлд байгаа бүх JSX файлуудыг багцалж `compiled` хавтас дотор нэг ширхэг ЖаваСкрипт багц (bundle) файл үүсгэнэ.
- `npm run build:w` - `npm run build` командтай адил, гэхдээ `--watch` горимоор [Webpack](#) багажийг ажиллуулна. Ингэснээр React компонентийг хянах, хэрэв тэдгээрийн аль нэгийг нь

өөрчилбөл дахин шинэчилж үүсгэдэг. Энэ нь хөгжүүлэлтийн үед хэрэгтэй байдаг бөгөөд өөрчлөгдсөн компонентоор шинээр багц үүсгэж веб хөтчийг дахин ачаалдаг (refresh). Үгүй бол та өөрчлөлт бүрийн дараа `npm run build` командыг үргэлж дуудаж ажиллуулж байх шаардлагатай болно. Ямар нэг юм хуучирсан тухай санамж `[DEP_WEBPACK_WATCH_WITHOUT_CALLBACK]` гарч болзошгүй, үүнийг алгасахад аюулгүй.

- `npm run lint` - ESLint багажийг төслийн бүх ЖаваСкрипт файлд ажиллуулна. Даалгавраар илгээх таны код чинь ESLint программыг ямар нэг санамжгүйгээр (warnings) давсан байх ёстой.

Бүх даалгаварыг гүйцэтгэсэн кодууд `project4` хавтас дотор байх ёстой.

Энэ төсөлд веб аппыг хөгжүүлэхэд түгээмэл болсон [ReactJS](#) фрэймворкийг ашиглана. Энэ төслийн зорилго бол таныг ReactJS-д хурдан суралцаж ICSI301 хичээлийн дараагийн төслүүдээр хийх веб аппыг хөгжүүлэхэд бэлэн болгох юм.

Бид веб аппдаа HTTP протоколоор хандахын тулд энгийн Node.js сервер ашиглана. Үүнийг `project4` хавтас дотроос дараах командаар асаана:

```
node webServer.js
```

`project4` хавтас доторх бүх файлуудад <http://localhost:3000> гэж эхлэх URL хаягаар хандана. <http://localhost:3000> линк дээр давхарлагдахгүй өөрийнхөө серверийг ажиллаж байгаа эсэхийг шалгаж болно. Энэ нь `index.html` файлыг таны веб хөтчид өгөх болно.

Та хөгжүүлэлтийн орчноо webpack багажийг watch горимоор тохируулахыг зөвлөж байна. Энэ нь та node веб сервер (`node webServer.js`) болон webpack (`npm run build:w`) программыг төслөө босгох болон туршихдаа ашиглаарай. Та тус тусад командын мөрөөр ажиллуулж болно. Ингэвэл Babel багажаар илрүүлсэн алдаа, түүний мэдэгдэл зэргийг webpack программын гаралтад харуулдаг учраас хэрэгтэй байх болно.

Хэрэв та MacOS зэрэг юникс-төст үйлдлийн системд ажиллаж байгаа бол команд нь:

```
node webServer.js & npm run build:w
```

Энэ нь нэг цонх дээр веб серверийг ард нь (background), webpack-ийг наана нь (foreground) харуулж ажиллуулдаг.

Виндоус үйлдлийн систем дээр веб серверийг ард нь, webpack-ийг наана нь ажиллуулахад Виндоусын хоёр команд нь:

```
start /B node webServer.js  
npm run build:w
```

Ард нь ажиллаж байгаа веб серверийг зогсоохын тулд дараах командыг ажиллуулна:

```
taskkill /IM node.exe /F
```

## Эхлэл

Энэ төслөөр таныг хичээлээр танилцсан model, view, controller (модель, харагдац, удирдлага) үлгэр загварыг ашиглах хэрэгтэй. Энэ загварын дагуу кодыг зохион байгуулах олон янзын арга бий. Бид нэгийг нь жишээ болгож ReactJS-ийн зарим үндсэн онцлог, мөн файл системд дэх байршлын зохиомж, модуль загвар зэргийг харуулж байна. Энэ үлгэр загварыг та дараагийн төслөөр хэрэгжүүлж яваарай.

Та энэ жишээг веб хөтөч дээрээ <http://localhost:3000/getting-started.html> URL хаягаар эхлүүлэн харж болно. Энэ хуудас нь ReactJS-ийг бодитоор ажиллуулж буй жишээ юм. `getting-started.html` хуудсанд байгаа HTML нь ReactJS аппыг зурж харуулахад зориулж нэг `<div>` элементийг, аппын ЖаваСкрипт багцыг `compiled/gettingStarted.bundle.js` холбож ажиллуулахад хэрэгтэй нэг script элементийг агуулж байгаа. webpack тохиргооны файлд `webpack.config.js` `gettingStarted.jsx` ReactJS файлаас үүсэх багцыг чиглүүлэн зааж өгсөн болно. `gettingStarted.jsx` бол нэг JSX программ бөгөөд ReactJS-ийн `Example` нэртэй компонентийг `getting-started.html` дотор байгаа нөгөө `<div>` рендерлэж харагдана.

Дахин ашиглах компонентийг хөгжүүлэхэд ReactJS компонент болон түүнд холбоотой CSS файлуудыг `components` дэд хавтас дотор зохион байгуулсан. Жишээ

нь, `Example` компонентийг `components/example/{Example.jsx,Example.css}` файлуудад хөгжүүлсэн.

Та `getting-started.html` хуудсыг дуудахад хэрэг болж байгаа файлуудыг (`getting-started.html`, `gettingStarted.jsx`, `components/example/Example.jsx`)

үзээрэй. Эдгээр нь ReactJS веб аппыг ажиллуулахад шаардлагатай ЖаваСкрипт болон JSX командуудыг тайлбар комментын хамт харуулж байгаа. Та хичээлээр босгох бусад компонентоо хөгжүүлэхэд энэ үлгэр загварыг нэрийн зохицолтойгоор (naming conventions) хэрэглэх ёстой.

Моделийн өгөгдөлийг ихэвчлэн веб серверээс авдаг. Веб сервер нь өгөгдлийн сангаас авдаг. Энэ төслөөр өгөгдлийн санг тохируулж хэрэглэхээс зайлсхийж дотоод файл системээс веб хөтчийн DOM-д шууд өгөгдлийг ачаалах HTML script тагийг бэлдэж өгсөн байгаа. Модел нь DOM-ийн `ICSI301models` нэртэй шинжид хадгалагдана. ReactJS компонент дотроос `window.ICSI301models` гэж шууд чөлөөтэй хандах боломжтой.

## Даалгавар 1: Example харагдацыг ойлгох ба шинэчлэх (5 оноо)

`getting-started.html` харагдац болон `Example` компонентийг үзэж ойлгох хэрэгтэй. Өөрийгөө ойлгосон гэдгийг харуулахын тулд дараах зүйлсийг хийнэ:

1. `Example` компонентийн модель өгөгдлийг "Unknown name" гэсний оронд өөрийн нэрээр шинэчлэнэ. "Unknown name" гэж хаана байгааг хайж болоод үүнийг солино.
2. `Example` компонентод байгаа `motto-update` гэсэн класстай `div` элементийн агуулгыг таны нэр болон богино (20 тэмдэгт хүртэл) уриа үг (motto) харуулдаг JSX командаар солино. Хэрэглэгчийн нэртэй адилхан уриа үгний утга нь моделийн өгөгдлөөс гарч ирэх ёстой. Үүнийг дэлгэцлэх зарим загвар `Example.css` файлд бичээрэй.

3. Өмнөх алхамаар хийсэн дэлгэцлэх хэсгийг хэрэглэгч өөрөө уриа үгийг шинэчилж болдог байхаар өргөтгөнө. Гарааны утга нь моделийн өгөгдлөөс гарч ирдэг байх ёстой.

## Даалгавар 2: Шинэ компонент үүсгэх - муж улсын харагдац (10 оноо)

Өгөгдсөн тэмдэгт мөрийг агуулж байгаа муж улсын бүх нэрсийг дэлгэцлэх шинэ компонент үүсгэнэ. Таны харагдац тэмдэгт мөр авдаг оролтын талбартай байх ёстой. Харагдац нь тухайн тэмдэгт мөрийг агуулж байгаа (том жижиг үсгийн ялгаагүй) муж улсын нэрсийг нэг **жагсаалтад** цагаан толгойн үсгийн дарааллаар дэлгэцлэнэ. Жишээ нь, тэмдэгт мөр нь "al" гэсэн утгатай байвал Alabama, Alaska, and California нэрс гарч ирнэ. Энэ хуудас муж улсын нэрсийг шүүхэд хэрэглэсэн тэмдэгт мөр утгыг бас дэлгэцлэдэг байна. Хэрэв тохирох муж улсын нэр олдохгүй бол веб хуудас нь холбогдох зурвасыг (юу ч харуулахгүй байхын оронд) харуулдаг байна. Тэмдэгт мөрийн утга хоосон бол бүх муж улсын нэрийг харуулдаг байна.

Даалгавар #1-д өгсөн шиг муж улсын нэрийг модель өгөгдөл болгож өгсөн байгаа. Үүнд дараах script тагаар холбож өгсөний дараа `window.ICSI301models.states` гэж хандаж болно:

```
<script src="modelData/states.js"></script>
```

`states.js` файлаас муж улсын нэрний өгөгдлийн форматын тайлбартай танилцаарай.

Энэ даалгаврыг эхлүүлэхэд туслах, чиглүүлэх зорилгоор `compiled/p2.bundle.js` файлыг холбож дуудсан `p2.html` файлыг нэрийн зохицолтойгоор бэлдэж өгсөн байгаа. Энэ ЖаваСкрипт багц файлыг `p2.jsx` файлаас webpack багажаар үүсгэсэн бөгөөд энэ нь `States` ReactJS компонентийг дэлгэцлэнэ. Энэ хуудсыг веб хөтөч дээр <http://localhost:3000/p2.html> URL хаягаар дуудаж үзээрэй.

Таны хөгжүүлэх файлууд бол:

- `components/states/States.jsx` - Муж улсын компонентийг хөгжүүлэх ReactJS компонент.
- `components/states/States.css` - Компонентод шаардлагатай дурын CSS загварууд. **Муж улсын нэрийн жагсаалтыг харуулахад зарим загваар хэрэгжүүлэх ёстой.**

## Даалгавар 3: Байршлыг өөрчлөх (5 оноо)

**Header** нэртэй ReactJS компонент үүсгэнэ. Энэ нь харагдацын дээд хэсэгт веб аппын толгойг (header) дэлгэцлэдэг. Толгойг энэ даалгаварт байгаа ReactJS-ийн бүх компонентод (`gettingStarted.jsx`, `p2.jsx`, `p4.jsx`, `p5.jsx`) нэмж өгнө. Өмнөх даалгаврын эхний хэсгийг (таны нэр ба уриа үгтэй) **өөрчлөхгүй**. Тэр хэсэг нь толгойгоос тусдаа байх ёстой. Өөрийн төсөөлөл, бүтээлч байдлаараа "зөвхөн чинийх" гэж харагдах толгойг үүсгээрэй. Энэ нь нэмэлт зураг, график, юунд таалагдана, тэр бүгдийг агуулж болно. Та компонентууд дахь JSX/ЖаваСкриптийг өргөтгөх бөгөөд гаднын ReactJS компонент, ЖаваСкрипт санг, жишээ нь, JQuery, ашиглаж болохгүй. Бүтээлч байгаарай!

Таны хөгжүүлэх файлууд бол:

- `components/header/Header.jsx` - Толгойг хөгжүүлэх ReactJS компонент. Энэ нь [React.Component](#) төрлийн Header классаар тодорхойлогдоно.
- `components/header/Header.css` - Өөрийн компонентод хэрэгтэй ямар нэг CSS загварууд. **Толгой компонентод заавал загвар тодорхойлж өгнө.**

Санамж: `getting-started.html` нь хуудасны дээд талдаа Даалгавар 3-аар хийсэн толгойг агуулж байх ёстой. Даалгавар 1.2-оор хийсэн уриа үгтэй хэсэг нь үүний яг доор нь байна. Бусад хуудасны харагдацууд (`p2.html`, `p4.html` and `p5.html`) энэ толгойг холбосон байна.

## Даалгавар 4: Харагдацуудыг динамикаар солих (10 оноо)

`p4.html` болон холбогдох JSX файлыг `p4.jsx` үүсгэнэ. Энэ нь хоёр харагдац компонентийг (`Example` болон `States` components) агуулна. `p4.jsx` нь энэ хоёр компонентийг хооронд нь сольж харуулдаг байх хэрэгтэй. Нэг харагдац нь дэлгэцлэгдсэн байхад нөгөөг нь сольж дэлгэцлэх нэг товч (`button`) байна. Жишээ нь, `States` дэлгэцлэгдсэн байхад үүний дээд талд байрлах товч нь "Switch to Example" гэж харагдах, `States` гэж дарахад өөрөө алга болоод `Example` дэлгэцлэгддэг байна.

Энэ даалгаврын хувьд дээрх файлуудыг үүсгэх хэрэгтэй. Мөн `webpack` тохиргооны `webpack.config.js` файлд `p4.html` хуудсанд ашиглах `compiled/p4.bundle.js` файлыг босгох тухай зааж өгөх хэрэгтэй болно. Хэрэв `webpack` багажийг `--watch` горимоор (i.e. `npm run build:w`) ажиллуулж байгаа бол та `webpack.config.js` файлаа өөрчилсний дараа дахин ажиллуулах шаардлагатай.

## Даалгавар 5: Нэг хуудаст апп (5 оноо)

Хэдийгээр Даалгавар 4-т гүйцэтгэсэн аргаар харагдац хооронд шилжиж болох ч энэ нь тухайн нэг харагдацыг заах URL-ийг хуваалцах (`share`) болон тэмдэглэж авах (`bookmark`) боломжийг олгохгүй. Веб хөтчийг дахин ачаалахад веб апп ямар хуудсыг дэлгэцэлж байсныг алдуулна.

Бид энэ дутагдлыг URL-д харагдацын мэдээллийг хадгалснаар шийдвэрлэх боломжтой. [React Router](#) нь ReactJS-д ийм функцээр хангаж өгдөг. Энэ даалгаврын хувьд `p4.html` файлаа `p5.html`, `p4.jsx` файлаа `p5.jsx` нэртэйгээр хуулбарлаж аваарай. Кодоо [React Router](#) модулийг ашиглан хоёр харагдацын хооронд нь сольж харуулдаг болгоно. Та **тодорхой загвартай хэрэгслийн-мөр маягийн удирдлагатай (`styled toolbar-like control`)** (энгийн бичвэр линк байх нь хангалтгүй) байх ёстой ба энэ нь хэрэглэгчид `example` болон `states` компонент харагдацыг сольж шилжих боломжийг олгодог байна.

Энэ бол бидний импортлож байгаа үндсэн ReactJS модулиас гаднах эхний өргөтгөл учир бид танд алхам нэг бүрийг нь тайлбарлаж өгье.

1. Төслийн `package.json` файлд `react-router` модулийг тодорхойлсон учир `npm install` командаар аль хэдийнэ татаад авчихсан байгаа. Бид үүнийг `p5.jsx` файл дотроо импортлож оруулах ёстой. Дараах мөрүүдийг нэмнэ:

```
import { HashRouter, Route, Link } from "react-router-dom"
;
```

Энэ мөр нь ЖаваСкрипт `import` командаар `React Router`-аас `ReactJS` компонентуудыг: `HashRouter`, `Route`, болон `Link` авч ирнэ. `React Router`-ийн `HashRouter` модуль `URL`-ийн ногдол хэсгийг (fragment portion) мэдээллийг хадгалах зорилгоор ашигладаг. Иймд бид `States` харагдацыг илэрхийлэх `p5.html#/states`, `Example` компонент харагдацыг тодорхойлох `p5.html#/example` тэмдгийг үүсгэж болно.

2. `React Router`-ийг ашигладаг хамгийн түгээмэл арга нь одоогийн `URL` дээр нь түшиглээд үзүүлэх гэж байгаа харагдацаа нөхцөлтэй рендерлэх юм. `Route` компонентийг `HashRouter` дотор оруулж ийм нөхцөлт рендерлэлтийг хэрэгжүүлж болдог:

```
<HashRouter>
  ...
  <Route path="/states" component={States} />
  <Route path="/example" component={Example} />
  ...
</HashRouter>
```

Хэрэв `URL`-д `#/states` агуулж байвал `States`, `#/example` гэж байвал `Example` компонентийг рендерлэнэ.

3. Хэдийгээр бид хайперлинк (жишээ нь, `<a>` tags) ашиглаад харагдацыг сольж болох боловч `react-`

router `Link` компонентоор хайперлинк үүсгэн ашиглахыг зөвлөдөг. Жишээ нь:

```
<Link to="/states">States</Link>
```

Дээрх код `href="#/states"` утгатай, "States" бичвэртэй хайперлинк үүсгэж өгнө.

## Загвар (5 оноо)

Энэ оноог лекц дээр хэлэлцсэн MVC загварын зөвлөмжийг дагаж зөв зохион байгуулсан тохиолдолд өгнө. **Таны код DOM-д шууд хандаж удирдсан байж болохгүй.** Нэмж хэлэхэд, таны код болон загвар цэвэрхэн, уншууртай байх ёстой. Илгээхээсээ өмнө ESLint багажийг ажиллуулсан байх хэрэгтэй. ESLint ямар нэг алдаа өгөөгүй байна.

## Хүлээлгэн өгөх үр дүн

**5 дахь хэсгийг хэрэгжүүлсний дараа 1-ээс 4 хүртэл ажиллаж байгааг нягтлаарай!**

[Даалгавар илгээх](#) зааврын дагуу аппаа бүхэлд нь (`project4` хавтас доторх бүгд) илгээнэ.

# Төсөл 5: Нэг хуудаст апп

Энэ төслөөр фото зураг хуваалцдаг вэб аппыг эхлэлийг хийхэд [ReactJS](#) фреймвөркийг [Material-UI](#) компоненттой хамтад нь ашиглана. Төслийн хоёрдахь хэсэгт өгөгдлийг серверээс авах талаар судлах болно.

## Тохиргоо

Та өөрийн систем дээрээ Node.js болон npm модуль менежерийг суулгасан байх ёстой. Хэрэв үгүй бол, [суулгах](#) зааврыг дагаж одоо суулгаарай.

`project5` нэрээр хавтас үүсгэж [энэ файлыг](#) дотор нь задалж хуулаарай. Энэ zip файл энэ даалгаварт ашиглах гарааны файлуудыг агуулж байгаа.

Энэ даалгаварт нэлээд олон node модулийг ашиглана. Жишээ нь, [Webpack](#), [Babel](#), [ESLint](#). Эдгээр багаж [ReactJS](#) веб апп босгох (build), мөн веб аппыг веб хөтчид өгч үйчлэх энгийн Node.js веб сервер ([ExpressJS](#)) үүсгэхэд хэрэг болно. Энэ нь бас [Material-UI](#) модулийг татаж авах бөгөөд энэ модуль React компонент болон CSS загварыг (stylesheet) агуулна. Эдгээр модулийг `project5` хавтас дотроос дараах командыг ажиллуулж суулгана:

```
npm install
```

[ReactJS](#) болон [Material-UI](#) компонентууд `node_modules` дэд хавтас дотор суух ч гэсэн үүнийг бид Node.js-ээр биш, веб хөтчид дуудаж ашиглах болно.

Өмнөх төслийн адил npm командыг татаж авсан эдгээр багажийг ажиллуулахад хэрэглэнэ. `package.json` файлд байгаа дараах npm скриптийг ажиллуулах боломжтой:

- `npm run lint` - ESLint багажийг төслийн бүх ЖаваСкрипт файлд ажиллуулна. Даалгавраар илгээх таны код чинь ESLint

программыг ямар нэг санамжгүйгээр (warnings) давсан байх ёстой.

- `npm run build` - [Webpack](#) багажийг `webpack.config.js` файлд байгаа тохиргоогоор ажиллуулж төсөлд байгаа бүх JSX файлуудыг багцалж `compiled` хавтас дотор нэг ширхэг ЖаваСкрипт багц (bundle) файл үүсгэнэ.
- `npm run build:w` - `npm run build` командтай адил, гэхдээ `--watch` горимоор [Webpack](#) багажийг ажиллуулна. Ингэснээр React компонентийг хянах, хэрэв тэдгээрийн аль нэгийг нь өөрчилбөл дахин шинэчилж үүсгэдэг.

Бүх даалгаварыг гүйцэтгэсэн кодууд `project5` хавтас дотор байх ёстой. Өмнөх төслөөр гүйцэтгэсэнтэй адил бидний өгч буй веб серверийг `project5` хавтас дотроос дараах командыг ажиллуулж асаана.

```
node webServer.js
```

Мөн өмнөхтэй адил дараах командаар:

```
node webServer.js & npm run build:w
```

командын мөрийн нэг цонх дээр веб сервер болон webpack программыг нэгэн зэрэг ажиллуулж болно.

## Даалгавар 1: Фото хуваалцдаг апп үүсгэх (40 оноо)

Таны үүсгэх PhotoApp-д зориулж бид суурь бүтцэт (skeleton) (`photoShare.jsx`-ийг дууддаг `photo-share.html`) гарааны кодыг өгсөн байгаа. Үүнийг <http://localhost:3000/photo-share.html> URL-ээр хандаж ажиллуулна. Энэ суурь код:

- Хичээлээр заасан мастер-деталь хэвийг хэрэгжүүлсэн [Material-UI](#) ашигладаг [ReactJS](#) веб аппыг дууддаг. Энэ апп дээд хэсэгтээ [Material-UI App Bar](#), зүүн хажуудаа `UserList`, контент

талбайдаа `UserDetail` эсвэл `UserPhotos` компонентийг байршуулсан байгаа.

- Нэг хуудаст аппд гүн холбоосыг (deep linking) бий болгохын тулд [React Router](#) компонентоор дараах гурван компонентийг заах замыг тохируулсан.
  1. `/users` нь `components/userList/` хавтас доторх `UserList` компонентийг заана.
  2. `/users/:userId` нь `components/userDetail/` хавтас доторх `UserDetail` компонентийг заана.
  3. `/photos/:userId` нь `components/userPhotos/` хавтас доторх `UserPhotos` компонентийг заана.

Дэлгэрэнгүйг `photoShare.jsx` дотор [HashRouter](#), болон [Route](#) компонентийг хэрхэн хэрэглэж байгаагаас харж болно. `components/*` хавтас дотор байгаа компонент бүрт хоосон CSS файл болон энгийн `render` функцийг юу хийх, модель өгөгдлийг хэрхэн хэрэглэх тухай тайлбарын хамт тодорхойлж өгсөн.

Энэ даалгаврын хувьд би өөрсдийн шидэт `ICSI301models` моделийг ашиглана. Энэ нь урьдаас бэлдсэн мэдээллийг дэлгэцлэхэд модель өгөгдөл болох юм. Өмнө нь ийм модельд `window.ICSI301Models` ашиглан хандаж байсныг санаж байгаа байх. Моделийн схемийг доор тайлбарласан болно.

Таны даалгавар бол өгсөн хоосон-программыг (skeleton) хуурмаг модель өгөгдөл дээр ажилладаг болгох юм. Энэ программд хэдийнэ `UserList`, `UserDetail`, болон `UserPhotos` зэрэг компонентийг холбож өгсөн байгаа. Энэ нь React Router компонентод шаардлагатай параметрууд холбож өгсөн бөгөөд компонентийг үүсгэх ихэнх ажлуудыг хийчихсэн. Эдгээрийг дараах шаардлагын дагуу хөгжүүлнэ:

- `components/userList` компонент системд байгаа бүх хэрэглэгчийн дэлгэрэнгүй мэдээллийг харуулахын тулд хэрэглэгчдийг жагсааж харуулдаг байна. Компонент нь хажуу хэсэгт (side bar) байрлах ба хэрэглэгчийн нэрсийн жагсаалтыг дэлгэцлэнэ. Хэрэглэгчийн нэр дээр дарахад

тухайн хэрэглэгчийн дэлгэрэнгүй мэдээллийг үндсэн хэсэгт (content area) дэлгэцэлдэг байна.

- `components/userDetail` компонентод React Router-ийн `props.match` шинжээр `userId`-ийг дамжуулна. Энэ харагдац нь хэрэглэгчийн дэлгэрэнгүйг харуулах бөгөөд тухайн хэрэглэгчийн оруулсан зургуудыг харуулах `UserPhotos` компонентруу заах линктэй байна.
- `components/userPhotos` компонентод `userId`-ийг дамжуулна. Сонгосон хэрэглэгчийн бүх зургийг харуулна. Тухайн хэрэглэгчтэй холбоотой бүх зургийг дэлгэцлэх ёстой. Зураг бүрийн хувьд бичсэн сэтгэгдлүүдийг харуулна. Сэтгэгдэл нэг бүрийн бичсэн огноо, тухайн сэтгэгдлийг бичсэн хэрэглэгч болон сэтгэгдлийн бичвэрийг харуулдаг байна. Тухайн сэтгэгдлийг бичсэн хэрэглэгчийн нэр нь линк байх ба дарахад тухайн хэрэглэгчийн дэлгэрэнгүй мэдээллийг үзүүлнэ.

Эдгээр компонентоос гадна `components/topBar` дотор байгаа `TopBar` компонентийг дараах шаардлагын дагуу шинэчлэнэ:

- `TopBar`-ын зүүн талд таны нэр байна.
- `TopBar`-ын баруун талд аппын агуулгыг үндсэн хэсэгт юу үзүүлж байгаагаас хамаарч гаргана. Жишээ нь, үндсэн хэсэгт хэрэглэгчийн дэлгэрэнгүй мэдээллийг харуулж байвал `TopBar` нь хэрэглэгчийн нэрийг, хэрэв хэрэглэгчийн зургийг үзүүлж байвал "Photos of " гээд араас нь хэрэглэгчийн нэрийг залгаж харуулдаг байна.

Энэ хоосон-программд React Router ашиглан аппын ялгаатай харагдац бүрт гүн-холбоос (deep-linking) үүсгэж өгсөн. Таны үүсгэх компонент энэ боломжийг эвдэхгүй байх ёстой. Энэ нь веб хөтчийг сэргээх (refresh) үед өмнө үзэж байснаараа гардаг байх боломжийг олгож байгаа юм. Бидний компонент үүсгэдэг стандарт арга нь гүн-холбоосыг автоматаар тохируулдаг. Компонентууд хооронд объектыг дамжуулахад үүнийг анхаарч байх ёстой. Хэрэв ямар нэг зүйл эвдсэн байвал харагдац бүр дээр сэргээх туршилт хийгээд үзээрэй.

Харагдах байдлын хувьд цаг их зарцуулах хэрэггүй, энэ нь аятайхан, ойлгомжтой байх ёстой. Мэдээллийг дэлгэцлэх зохиомж нь цэвэр байна (Жишээ нь, зураг бүрийн хувьд холбогдох коментууд нь ойлгомжтой байх).

## Фото апп модель өгөгдөл

Энэ даалгаврын хувьд бид өмнөх даалгавраар ашигласан, нөгөө DOM-д ачаалсан шидэт модель өгөгдлийг ашиглана. Модель нь дөрвөн төрлийн объектойд. Эдгээр нь `user`, `photo`, `comment`, болон `SchemaInfo` төрөл.

- Фото хуваалцдаг сайт дээрх фотонуудыг хэрэглэгчээр нь зохион байгуулсан. Бид хэрэглэгчийг `user` объектоор илэрхийлэх ба дараах шинжүүдтэй:

|                            |                                    |
|----------------------------|------------------------------------|
| <code>_id</code> :         | Хэрэглэгчийн ID.                   |
| <code>first_name</code> :  | Хэрэглэгчийн өөрийн нэр.           |
| <code>last_name</code> :   | Хэрэглэгчийн овог.                 |
| <code>location</code> :    | Хэрэглэгчийн байршил.              |
| <code>description</code> : | Хэрэглэгчийн талаарх товч тайлбар. |
| <code>occupation</code> :  | Хэрэглэгчийн ажил мэргэжил.        |

- `window.ICS1301models userModel(user_id)` DOM функц нь хэрэглэгчийн `user_id` ID-аар `user` объектыг буцаадаг. Харин `window.ICS1301models userListModel()` DOM функц нь аппын бүх хэрэглэгчийн `user` объектыг буцаадаг.
- Хэрэглэгч бүр олон фото серверт хуулах боломжтой. Хэрэглэгчийн нэг зургийг `photo` объектоор, дараах шинжтэйгээр илэрхийлдэг:

|                          |                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------|
| <code>_id</code> :       | Фото зургийн ID.                                                                     |
| <code>user_id</code> :   | Фото зургийг үүсгэсэн хэрэглэгчийн <code>user</code> ID.                             |
| <code>date_time</code> : | Фото зургийг өгөгдлийн санд нэмсэн огноо ба цаг.                                     |
| <code>file_name</code> : | Фото зургийг агуулж байгаа файлын нэр ( <code>project5/images</code> хавтас доторх). |
| <code>comments</code> :  | Фото зургийн бүх сэтгэгдлийг <code>comments</code> хадгалсан обьектон                |

хүснэгтэн хувьсагч, Нэг `comment` нь тухайн фото зураг дээр бичигдсэн сэтгэгдэл.

- `window.ICSI301models.photoOfUserModel(user_id)` DOM функц нь хэрэглэгчийн `user_id`-д холбогдох `photo` объектуудыг агуулсан хүснэгтэн хувьсагчийг буцаадаг.
- Фото бүр олон сэтгэгдэлтэй байж болно (ямар ч хэрэглэгч ямар ч фото зурагт сэтгэгдэл бичиж болно). Сэтгэгдэл `comment` объект дараах шинжүүдтэй:

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| <code>_id</code> :       | Сэтгэгдлийн ID.                                         |
| <code>photo_id</code> :  | Сэтгэгдэл харгалзах <code>photo</code> фото зургийн ID. |
| <code>user</code> :      | Сэтгэгдэл бичсэн хэрэглэгчийн <code>user</code> объект. |
| <code>date_time</code> : | Сэтгэгдлийг бичсэн огноо ба цаг.                        |
| <code>comment</code> :   | Сэтгэгдлийн бичвэр.                                     |

- Турших зорилгоор `SchemaInfo` объектыг дараах шинжээр үүсгэсэн:

|                               |                                                                      |
|-------------------------------|----------------------------------------------------------------------|
| <code>_id</code> :            | <code>SchemaInfo</code> схемийн ID.                                  |
| <code>__v</code> :            | <code>SchemaInfo</code> объектын хувилбарын дугаар.                  |
| <code>load_date_time</code> : | <code>SchemaInfo</code> ачаалах үеийн огноо ба цаг. Нэг тэмдэгт мөр. |

## Даалгавар 2: Веб серверээс моделийн өгөгдлийг авах (20 оноо)

Даалгавар 1-ийг гүйцэтгэсэний дараа манай фото аппын front-end бодит апп шиг харагдах ёстой. Гол асуудал нь ЖаваСкрипт болон DOM-ыг ашиглаад хуурмаг модель өгөгдлийг ачаалсан байсан. Даалгавар 2-ын хувьд, бид энэ арга залиг хаяад бодит апптай ижил серверээс модель өгөгдөл авдаг болгоно.

Энэ төслөөр өгсөн `webServer.js` нь Даалгавар 1-д DOM-д ачаалж ашигласан `ICSI301Models`-ийг `ExpressJS` чиглүүлэгч (route) ашиглан серверээс өгөгдөл авах боломжийг олгож байгаа. Энэ API-ыг `webServer.js`-ээр гаргаж байгаа ба харгалзах URL хаягаар HTTP GET хүсэлт явуулахад `ICSI301Models` моделиудаас өгөгдлийг буцааж

өгдөг. Эдгээр GET хүсэлтэд хариулах HTTP хариулт нь JSON форматтай байна. Энэ API бол:

- `/test/info` - `ICSI301models.schemaInfo()` объектыг буцаана. Энэ нь URL нь модель дуудаж турших зориулалттай.
- `/user/list` - `ICSI301models.userListModel()` функцийг дуудаж буцаана..
- `/user/:id` - `ICSI301models.userModel(id)` функцийг дуудаж буцаана.
- `/photosOfUser/:id` - `ICSI301models.photoOfUserModel(id)` функцийг дуудаж буцаана..

Дээрх URL-ийг веб хөтөчийн хаягийн мөр дээрээ бичээд бодитоор хэрхэн ажиллаж байгааг харж болно. Жишээ нь, <http://localhost:3000/test/info> болон <http://localhost:3000/user/list> линк нь веб хөтчийн цонх дээр моделийн өгөгдлийг JSON форматаар харуулна.

Веб сервер дээрээс моделийг авдаг болгож программаа хөрвүүлэхийн тулд `lib/fetchModelData.js` файлд байгаа `fetchModel` функцийг хөгжүүлэх ёстой. Энэ функцийг дараах тодорхойлолтын дагуу хөгжүүлнэ:

```
/**
 * fetchModel - Веб серверээс моделийг авдаг
 * @param {string} url      GET хүсэлт явуулах URL
 *
 * @returns GET хүсэлтийн хариултад ирэх JSON объектоор
 * дүүргэсэн Promise буцаана.
 *
 * Өгөгдөл нь нэг объектын "data" нэртэй шинжэд байна.
 *
 * Хэрэв хүсэлт явуулахад алдаа гарвал Promise нь дараах
 * шинжүүдийг агуулсан объектыг буцаана.
 *
 * {number} status      HTTP хариултын төлөв
 * {string} statusText   xhr хүсэлтээр ирсэн өгөгдөл
 */
```

Хэдийгээр энэ функцийг олон янзын бэлэн хөгжүүлсэн модуль ашиглан хийж болох ч та AJAX-ийн доод түвшний мэдлэгт суралцах хэрэгтэй. Та `fetchModel` функцийг кодлохдоо өөр бусад санг ашиглахгүй байх; XMLHttpRequest DOM объектыг үүсгэх болон түүний үзэгдлийг барьж авч хэрэглэх ЖаваСкрипт кодыг бичээрэй.

Таны код олон `fetchModel` хүсэлтүүдийг хүлээн авч хариулдаг байх ёстой. Өөрийн `fetchModel` функцийг ажиллаж байгаа эсэхийг харахын тулд <http://localhost:3000/photo-share.html> хаягаар орж үзээрэй. Энэ нь <http://localhost:3000/test/info> хаягруу хүсэлт явуулаад буцаж ирэх хувилбарын дугаарыг дэлгэцэлдэг юм. Хувилбарын дугаарыг өөрийн аппынхаа `TopBar` компонентод дэлгэцэлдэг байна.

`lib/fetchModelData.js` файлд `fetchModel` функцийг амжилттай кодлосны дараа та дараах файлуудын кодыг өөрчлөөрэй.

- `components/userDetail/UserDetail.jsx`
- `components/userList/UserList.jsx`
- `components/userPhotos/UserPhotos.jsx`

Дээрх `fetchModel` функц нь серверээс өгөгдлийг авах хүсэлт тавьдаг. Таны кодоод `window.ICSI301models` хувьсагчийг шууд дуудаж ашиглах код байх ёсгүй бөгөөд таны аппын `photo-share.html` файлд дараах мөр байх ёсгүй:

```
<script src="modelData/photoApp.js"></script>
```

## Загвар (5 оноо)

Энэ оноог MVC загварын зөвлөмжийг дагаж зөв зохион байгуулсан тохиолдолд өгнө. Нэмж хэлэхэд, таны код болон компонент цэвэрхэн, уншууртай байх ёстой ба таны апп ядаж "тодорхой хэмжээнд гоё" харагддаг, хэрэглэгддэг байна.

Бид [Material-UI](#) гэх Гүүглээс гаргасан [Material Design](#) зөвлөмжийг хэрэгжүүлсэн React компонентийг ашиглана. Хичээл дээр ярьсан Master-Detail загварыг хэрэгжүүлэхэд Material UI-ийн [Grid component](#) ашиглаж харуулсан. Мөн [App Bar](#) компонентоор аппын толгой хэсгийг хийсэн байгаа. Хэдийгээр Material Design загварт

бүрэн нийцсэн апп хөгжүүлэхийг шаардахгүй ч боломжтой бол [Material-UI](#) элементүүдийг аль болох ашиглаж хийгээрэй.

Бас илгээхээсээ өмнө ESLint ажиллуулахаа мартаузай. ESLint ямар нэг алдаа өгөхгүй ёсгүй.

## Нэмэлт оноо (5 оноо)

`userPhotos` компонент нь хэрэглэгчийн бүх зургаас гадна зурагт бичсэн коментуудыг харуулах ёстой. Энэ арга нь маш олон зурагтай хэрэглэгчийн хувьд тийм ч сайн ажиллахгүй. Нэмэлт онооны даалгавраар нэг зургийг коменттой нь харуулдаг болгож хөгжүүлэх ба өмнөх болон дараах гэсэн механизмыг (жишээ нь, stepper буюу алхамт байдлаар) хэрэгжүүлж хэрэглэгчийн зургийг дэлгэцэлдэг байна.

Энэ даалгаврт нэмэлт оноо авахын тулд таны шийдэл дараах шиг байна:

- Фото аппдаа "Нэмэлт сонголт" ойлголтыг оруулж ирнэ. Апп эхлэж ажиллахад энэ сонголт нь үргэлж идэвхгүй байна. `TopBar` компонент "Нэмэлт сонголтыг" идэвхжүүлэх чагтлах товлууртай (checkbox) байх үүнийг чагталсан тохиолдолд "Нэмэлт сонголт" идэвхтэй болно. Эсрэг тохиолдолд идэвхгүй болно. Чагтлах товчлуур дарснаар сонголтыг идэвхтэй/идэвхгүй болгоно.
- Таны апп "Нэмэлт сонголт"-ыг идэвхжүүлээгүй бол зураг үздэг ердийн харагдацаа ашигладаг байх юм. Хэрэв идэвхжсэн бол зургуудыг нэг нэгээр нь алхамт байдлаар харуулдаг байна.
- Алхамаар харуулах хэрэглэгчийн интерфэйс нь хэрэглэгч зургуудын хамгийн эхнийхийг үзэж байвал бол өмнөх зурагруу, хамгийн сүүлд очсон бол дараах зурагруу шилжих боломжгүй учир үүнийг маш ойлгомжтой (жишээ нь, товчлуурыг идэвхгүй болгох) харуулдаг байна.
- Таны апп тухайн нэг зургийн URL хаягийг веб хөтчийн хаягийн мөрөөс хуулж шэйрлэх болон тэмдэглэж авах

(bookmark) боломжийг олгодог байна. Веб хөтчийн өмнөх болон дараах шилжих товч нь өмнөхрүү буцах, эргэж буцах гээд хийдэг үйлдлээ хийдэг байна. Тухайн зургийг заах гүн холбоосоор аппруу орох үед алхамт байдлаар харуулах үүргээрээ ажилладаг байна.

**Санамж:** Энэ нэмэлт онооны даалгаврыг хийх нь төслөөр хийсэн нэмэлт онооны бус бусад даалгавруудын хэд хэдэн хэсэгт өөрчлөх шаардлага гарна. *Онцлог чагтлах (feature flag)* товчлуураар ийм боломжийг хийх нь веб аппликейшнд нийтлэг практик байдаг. Гэхдээ нэмэлт онооны бус бусад даалгавруудаа эвдэж нэмэлт оноо авах гэж байгад илүү их оноо алдах эрсдэл бий. Болгоомжтой байгаарай.

## Хүлээлгэн өгөх үр дүн

Ердийн [Даалгавар илгээх](#) зааврын дагуу аппаа бүхэлд нь (project4 хавтас доторх бүгд) илгээнэ. Илгээхээсээ өмнө зааврын дагуу төслийн хавтасаа цэвэрлээрэй. Ялангуяа, 180MB хэмжээтэй node\_modules/ хавтсыг дэмий хадгалахыг хүсэхгүй байна. compiled/ хавтсаа ч бас устгахаа мартуузай.

# Төсөл 6: Аппсервер ба өгөгдлийн сан

Энэ төсөлд та өгөгдлийн сангийн системийг эхлүүлж, 5-р төсөлд хийсэн өөрийн фото зураг хуваалцдаг веб аппыг харагдацад (view) моделийг өгөгдлийн сангаас авдаг байхаар хөгжүүлнэ. Бид Project 5 дээр ашигласантай ижил, гэхдээ шинэ вэб сервер `webServer.js` интерфэйсийг өгсөн байгаа. Энэ нь өгөгдлийн сантай холбогддог юм. Энэ нь танд өөрийн аппыг албан ёсоор full stack программ болгох боломжийг олгоно.

## Тохиргоо

Та өөрийн систем дээр Mongo.DB болон Node.js татсан байх шаардлагатай. Хэрвээ татаагүй бол [суулгах](#) зааврыг дагаж одоо суулгаарай.

### \*САНАМЖ!\*

Төсөл 6-ийн тохиргоо нь бусад төслүүдээс ялгаатай. Та `project6` нэртэй хавтас үүсгээд `project5`-ын файлуудаа хуулаарай. `project6` хавтаст энэ `zip file` файлыг дотор нь задлаарай. Энэхүү `zip` файл нь `package.json`, `webServer.js`, `.eslintrc.json`, болон `index.html` гэх мэт файлуудыг дарж хуулах бөгөөд хэд хэдэн шинэ файл, хавтсуудыг нэмэх болно. Хэрэв та өөрийн `project5` хавтас доторх эдгээр файлуудын аль нэгэнд шаардлагатай өөрчлөлтийг хийсэн, гэхдээ магадлал бага байх, тохиолдолд задлах үйлдлийг хийсний дараа өөрийн хийсэн өөрчлөлтийг дахин хийх шаардлагатай болно.

Төсөл 6 файлуудтай болсны дараа дараах командыг ашиглан шаардлагатай программыг татаж авна уу:

```
npm install
```

Хэрэв та `npm install` командыг `node_modules` хавтсыг арилгахгүйгээр ажилуулбал дараах анхааруулгатай тулгарч магадгүй:

```
npm WARN library_A requires a peer of library_B but none is installed. You must install peer dependencies yourself.
```

Та бидний ашигладаг хамаарлууд нь реер хамаарлын шаардлагатай таарахгүй байх боломжтой учраас дараах анхааруулгатай тулгарч магадгүй юм.

Энэ болон бусад хичээлийн даалгаврын хувьд бид вэб программын бүх гурван шатлалууд болох хөтөч, вэб сервер, мэдээллийн санг таны дотоод машин дээр ажиллуулна.

## MongoDB өгөгдлийн санг эхлүүлэх

Суулгах [зааварт](#) заасны дагуу өөрийн орчинд MongoDB ажиллаж байгаа эсэхийг шалгаарай. . Өгөгдлийн сан хаагдах хүртэл `mongod` команд буцаагдахгүй тул та үүнийг тусдаа цонхонд эсвэл арын процесс болгон ажиллуулахыг хүсэх болно. (Жишээ нь: `mongod (args)` & Linux/MacOS дээр).

MongoDB серверийг ажиллуулсны дараа та дараах командыг ажиллуулж зургийн программын өгөгдлийг ажиллуулж болно:

```
node loadDatabase.js
```

Энэ программ нь өмнөх төслүүдийн (жишээ нь: `modelData/photoApp.js`) гэх мэт хуурамч загварын өгөгдлийг мэдээллийн санд ачаалуулдаг. Манай апп-д ямар нэгэн зүйл нэмэх, шинэчлэх ямар ч шаардлага байхгүй тул та `loadDatabase.js` -г нэг удаа ажиллуулахад л хангалттай. Программ нь өгөгдлийн багцыг ачаалалхаас өмнө өгөгдлийн санд байгаа бүх зүйлийг устгадаг тул олон удаа ажиллуулахад аюулгүй байдаг.

Бид [MongooseJS](#) Object Definition Language (ODL)-ийг ашиглан MongoDB-д photo app өгөгдлийг хадгалах `schema` -ийг тодорхойлно. Схемийн тодорхойлсон файлууд нь лавлах(directory) `schema`-д байна:

- `schema/user.js` - Хэрэглэгч бүрийг дүрсэлсэн объектуудыг агуулсан Хэрэглэгчийн цуглуулгыг тодорхойлно.

- `schema/photo.js` - Зураг бүрийг дүрсэлсэн объектуудыг агуулсан Photos цуглуулгыг тодорхойлно. Энэ нь мөн зураг дээр бичсэн сэтгэгдлийг хадгалахад ашигладаг объектуудыг тодорхойлдог.
- `schema/schemaInfo.js` - Схемийн хувилбарыг тодорхойлсон объектыг агуулсан SchemaInfo цуглуулгыг тодорхойлно.

Эдгээр файлуудыг өгөгдлийн сан үүсгэхэд ашигладаг `loadDatabase.js` программ болон мэдээллийн санд хандахад ашигладаг `webServer.js` хоёуланг нь ачаалдаг. Тайлбар: Өгөгдлийн санд хадгалагдсан объектын схем нь өмнөх даалгаварт ашигласан `ICSI301models` JavaScript объектуудтай төстэй боловч зайлшгүй ялгаатай.

## Node.js вэб серверийг эхлүүлэх

Мэдээллийн санг ажиллуулсны дараа вэб серверээ эхлүүлэх хэрэгтэй болно. Үүнийг өмнөх даалгавартай ижил тушаалаар хийж болно (жишээ нь: `node webServer.js`). Вэб серверээ `project6` directory ашиглан command хийж эхлүүлээрэй:

```
node webServer.js
```

Хэрэв та дээрх командыг ашигладаг бол **серверийн кодыг өөрчлөх болгондоо вэб серверээ дахин эхлүүлэхээ мартуузай**. Та мөн серверийн кодын өөрчлөлтийг ажиглаж , вэб серверийг автоматаар дахин эхлүүлэх `nodemon`-г ажиглаж болно:

```
nodemon webServer.js
```

Хэрэв Nodemon-ийг ажиллуулах үед таний дэлгэц дээр `command not found: nodemon` гэсэн алдаа гарвал энийг ажиллуулж үзнэ үү

```
node_modules/.bin/nodemon webServer.js
```

Project 6-ийн шинэ файлуудаар Photo Share программаа шинэчилж, мэдээллийн сан болон вэб серверийг ажиллуулсны дараа даалгавраа

үргэлжлүүлэхийн өмнө программ ажиллаж байгаа эсэхийг шалгана уу.

## Өгөгдлийн сан ашигладаг веб сервер болгон хөрвүүлэх (40 оноо)

Энэ төсөлд бидний танд өгөх `webServer.js` Төслийн 5 `webServer.js` - тэй адил бөгөөд программын загвар татах маршрутууд нь мэдээллийн сангаас илүү шидэт `ICSI301models` загваруудыг ашигладаг. Таны даалгавар бол MongoDB мэдээллийн санг ашиглан бүх маршрутыг хөрвүүлэх явдал юм. Таны кодоод `ICSI301models`-д хандах эрх байх ёсгүй бөгөөд таны программ дараах мөргүйгээр ажиллах ёстой:

```
const ICSI301models = require('./modelData/photoApp.js').ICSI301models;
```

`webServer.js` дээр `webServer.js` доторх `console.log` нь Browser дээр биш терминал дээр хэвлэгдэх болно гэдгийг анхаарна уу.

### Web Server API

Төсөл 5-ын нэгэн адил вэб сервер нь тодорхой URL-ууд руу HTTP GET хүсэлтийн хариуд JSON кодлогдсон загварын өгөгдлийг буцаана. Бид ямар URL-уудыг дэмжих, юуг буцаах талаар дараах тодорхойлолтыг өгөв. Таны вэб сервер дараах загвар татах API-г дэмжих ёстой:

- `/test` - Өгөгдлийн сангийн схемийн мэдээлэл (`/test/info`) болон объектын тоо (`/test/counts`) -ийг буцаана. Энэхүү интерфэйс нь туршилтанд зориулагдсан бөгөөд бид танд жишээ болгон мэдээллийн сангаас мэдээлэл татах программыг санал болгож байна. Та үүнийг өөрчлөх шаардлагагүй болно.
- `/user/list` - Навигацийн хажуугийн жагсаалтад тохирох хэрэглэгчдийн загваруудын жагсаалтыг буцаана. Бид олон тооны хэрэглэгчдийг хүлээж байгаа тул энэ API нь зөвхөн наavigацийн хажуугийн самбарт шаардлагатай хэрэглэгчийн шинж чанаруудын массивыг буцаана (`_id`, `first_name`,

`last_name`). Энэ нь өгсөн код дахь `ICSI301models.userListModel()` дуудлагыг орлоно.

- `/user/:id` - `id`-ийн `_id`-тэй хэрэглэгчийн дэлгэрэнгүй мэдээллийг буцаана. Энэ нь хэрэглэгчийн талаарх дэлгэрэнгүй мэдээллийг (`_id`, `first_name`, `last_name`, `location`, `description`, `occupation`) буцааж өгөх ба `ICSI301models userModel()` дуудлагыг солих ёстой. Хэрэв хэрэглэгчийн `id`-ээс өөр зүйл өгсөн бол хариу нь HTTP статус 400 ба мэдээллийн мессеж байх ёстой.
- `/photosOfUser/:id` - `:id` - `id`-ийн `_id`-тэй хэрэглэгчийн зургийг буцаана. Энэ дуудлага нь хэрэглэгчийн бүх зураг, мөн зураг дээрх сэтгэгдлүүд зэрэг зураг үзэхэд шаардлагатай бүх загварын өгөгдлийг үүсгэдэг. Зургийн шинж чанарууд нь (`_id`, `user_id`, `comments`, `file_name`, `date_time`) байх ба тайлбарын массивын элементүүд (`comment`, `date_time`, `_id`, `user`) байх ёстой бөгөөд зөвхөн хамгийн бага хэрэглэгч-ийн объектын мэдээлэл (`_id`, `first_name`, `last_name`) байх ёстой. Энэ нь `ICSI301models.photoOfUserModel()` дуудлагыг орлоно. Хэрэв хэрэглэгчийн `id`-ээс өөр зүйл өгсөн бол хариу нь HTTP статус 400 ба мэдээллийн мессеж байх ёстой. Энэ API-д мэдээллийн сан дахь олон өөр объектоос угсрах шаардлагатай болохыг анхаарна уу. `Mongoose`-ээс өгсөн мэдээллийн баазын асинхрон интерфэйс нь эдгээр олон объект татах үйлдлийг нэгэн зэрэг хийх боломжтой гэсэн үг юм. Таны код татан авалтыг шаардлагагүй цуваа болгох ёсгүй. Даалгаврын `package.json` файл нь `async` модулийг татаж авдаг бөгөөд энэ нь зэрэгцээ таталтыг удирдахад тустай.

Таны вэб сервер зохих API-д нийцэж байгаа эсэхийг шалгахын тулд бид дэд лавлах `test` багцыг санал болгож байна. **Илгээхээсээ өмнө багц дахь бүх шалгалтыг давсан эсэхийг шалгана** уу. Дэлгэрэнгүйг доорх Туршилтын хэсгээс үзнэ үү.

Таны GET хүсэлтүүд нь `ICSI301models` функцүүдийн буцаадаг зүйлтэй яг ижил зүйлийг буцаадаггүй боловч харагдац бүрийн загварын өгөгдлийг нэг `fetchModel` дуудлагаар харуулахын тулд таны апп-д шаардлагатай мэдээллийг буцаах шаардлагатай. Та UI-ийн хэрэгцээг

хангахын тулд өөрийн хариултыг бий болгохын тулд мэдээллийн сангаас ирж буй объектуудыг дэд тохиргоо болон/эсвэл нэмэгдүүлэх шаардлагатай болно. Энэ даалгаврын хувьд та мэдээллийн сангийн схемийг ямар нэгэн байдлаар өөрчлөх эрхгүй.

### **\*ЧУХАЛ!\***

Эдгээр Экспресс хүсэлт боловсруулагчийг хэрэгжүүлэхийн тулд хоёр өөр "загвар" өгөгдлийн объекттой харилцах шаардлагатай. Mongoose систем нь MongoDB-д хадгалагдсан объектуудаас **models**-уудыг буцаадаг бол хүсэлт нь өөрөө Photo App харагдацад шаардлагатай өгөгдлийн загваруудыг буцаах ёстой. Харамсалтай нь, Mongoose загварууд нь мэдээллийн сангийн схемээр, урд талын загварууд нь UI-ийн харагдацын шаардлагаар тохируулагддаг тул тэдгээр нь төгс нийцдэггүй. Эдгээр хүсэлтийг шийдвэрлэхийн тулд мэдээллийн сангаас буцаасан Mongoose загваруудаас урд хэсэгт шаардлагатай загварыг угсрахын тулд боловсруулах шаардлагатай болно.

Mongoose-ийн буцаасан загварууд нь JavaScript объектууд боловч тэдгээрт тусгай боловсруулалт хийгдсэн тул зарлагдсан схемд тохирохгүй аливаа өөрчлөлтийг хаях тул энэ боловсруулалтыг хийхдээ болгоомжтой байх хэрэгтэй. Энэ нь Mongoose загварыг урд талын хүлээгдэж буй шинж чанаруудтай болгохын тулд зүгээр л шинэчлэх нь санаснаар болохгүй гэсэн үг юм.

Энэ асуудлыг шийдэх нэг арга бол шинэ JavaScript объект үүсгэж, Mongoose загвараас авсан талбаруудаар дүүргэх явдал юм. API-д заасан яг JavaScript өгөгдлийн бүтцийг ийм байдлаар бүтээж, HTTP хариултанд буцааж өгч болно. API загвар нь Mongoose загвартай нягт таарч байгаа REST эндпойнтыг хувьд өөр нэг арга бол Mongoose проекцын операторыг (**сонгож**) ашиглан хайлт нь зөвхөн API-ийн заасан загварын талбаруудыг буцаана. Хэсэг нь эдгээр техникийг илүү нарийвчлан авч үзэх болно.

Mongoose болон API загварууд ижил байсан ч проекц хийхгүйгээр Mongoose загварыг буцаах нь программчлалын муу практик юм. Ирээдүйн төслүүд өгөгдлийн сангийн схемийг өргөтгөх болно (жишээ нь, хэрэглэгчийн загварт нууц үг нэмэх), гэхдээ бид эдгээр шинж чанаруудыг API загварт оруулахыг хүсэхгүй байна.

## Даалгавар 2: Өөрийн программыг `axios` ашигладаг болгох (10 оноо)

Бид REST API-г илүү ашиглах дараагийн даалгаварт бэлтгэхийн тулд вэб серверээс загваруудыг татахын тулд өөрийн `fetchModel` горимоос илүү `axios` ашиглахаар зургийн программаа хөрвүүлээрэй. `Axios` бол өгөгдөл татах боломжтой олон npm багцуудын нэг юм. Энэ нь манай `fetchModel` функцээс илүү хурдан бөгөөд илүү олон функцтэй бөгөөд бид үүнийг цаашид хүсэлтээ гаргахад ашиглах болно. Энэ асуудлыг дуусгасны дараа таны зургийн апп `fetchModel` функцийг тодорхойлолтгүйгээр ажиллах ёстой. `fetchModel` функцийг тодорхойлолтыг арилгахын өмнө болон дараа нь таны программын ажиллагаа яг ижил байх ёстой -- энэ шилжүүлгийг хийх явцад ямар нэгэн зүйл эвдэхгүй байгаа эсэхийг шалгаарай.

Ялангуяа та `photoShare.jsx` эсвэл бусад бүрэлдэхүүн хэсгүүдэд `fetchModel` руу залгах бүртээ үүнийг `axios.get` руу залгах хэрэгтэй.. Дэлгэрэнгүй мэдээллийг `axios` баримтаас үзнэ үү.

`axios` руу шилжих зарим зөвлөмжийг энд оруулав:

- `axios` ашиглан өгөгдөл татаж авах аливаа бүрэлдэхүүн хэсэгт дараах байдлаар импортлохоо мартуузай:

```
import axios from 'axios';
```

- `axios.get` нь `fetchModel` шиг амлалт буцаадаг болохыг анхаарна уу.. Бид `.then` болон `.catch` тус тус ашиглан амжилт ба алдаа зохицуулагчийг хавсаргаж болно. Аливаа хүсэлтийн амжилт болон алдааны тохиолдлуудыг хоёуланг нь зохицуулах нь чухал юм. `axios.get` -д хавсаргахдаа `fetchModel`-аас зохицуулагчийн логикийг маш их эсвэл огт өөрчлөх шаардлагагүй.

## Тест

Лавлах тестийн хувьд бид таны программын зөвхөн арын хэсгийн `test` өгдөг. Тест нь Тест нь Node.js тест бичихэд зориулагдсан түгээмэл хүрээ болох `Mocha` ашигладаг. Туршилтын орчныг тохируулахын тулд `test`-ийн дэд директор дотроос `Mocha` болон холбогдох бүх хамаарлыг татахын тулд `npm install` хийгээрэй. Үүнийг хийснийхээ дараа та тестийн лавлах доторх тушаалыг ажиллуулж тестийг ажиллуулж болно: `npm test`.

`npm test` `Mocha` хэл дээр бичигдсэн программ (жишээ нь, `describe()` and `it()`) гэсэн гурван Photo App backend URL (`/user/list,` `/user/ID,` `/photosOfUser/ID`)-ийг шалгадаг `test/serverApiTest.js` файлыг ажиллуулдаг. **Арын чиглүүлэлтийн функцууд нь техникийн үзүүлэлтэд нийцэж байгаа гэдэгт итгэлтэй байхын тулд илгээхээсээ өмнө бидний өгсөн бүх туршилтыг давсан эсэхийг шалгана уу. Таны төслийн үнэлгээний нэг хэсэг нь эдгээр шалгалтын хэд нь тэнцсэнээс хамаарна.**

## Нэмэлт оноо (10 оноо)

Таны Photo App-ын маркетингийн хэлтэс нь программыг нийгмийн сүлжээнд илүү ээлтэй болгохын тулд "жигжиг" засвар хийсэн. Өөрчлөлт нь:

- Хэрэглэгчдийн жагсаалтыг агуулсан хажуугийн навигацийн талбарт хэрэглэгчийн нэр бүрийн хажууд тоолох хоёр `count bubbles` байх ёстой. Эхний тооны бөмбөлөг (ногоон өнгөтэй) нь хэрэглэгчийн системд байгаа зургийн тоо байх ёстой. Хоёр дахь бөмбөлөг (улаан өнгөтэй) нь хэрэглэгчийн бичсэн сэтгэгдлүүдийн тоо байх ёстой.
- Хэрэглэгчийн сэтгэгдлийг тоолох хөөс дээр дарснаар хэрэглэгчийн бүх сэтгэгдлийг харуулдаг шинэ харах бүрэлдэхүүн хэсэг рүү очно. Хэрэглэгчийн сэтгэгдэл бүрийн хувьд сэтгэгдэл бичсэн зургийн жигжиг өнгөц зураг болон сэтгэгдлийн текстийг харуулах ёстой. Сэтгэгдэл эсвэл зураг дээр дарснаар тухайн зураг болон түүний бүх тайлбарыг

агуулсан зургийн дэлгэрэнгүй харагдац руу шилжих хэрэгтэй. Та 5-р төсөлд `stepper` нэмэлт кредитийг хэрэгжүүлсэн эсэхээс яг тодорхой харагдах байдал хамаарна.

- Төслийн 5-ын дэвшилтэт функцийн туг идэвхжсэн тохиолдолд л энэ өөрчлөлт харагдах болно. Хэрэв та Төслийн 5-ын нэмэлт кредитийг хийгээгүй бол алхам алхамыг хийх шаардлагагүй, гэхдээ дээрх нэмэлт зээлийн функцийг асаах, унтраах боломжтой болгохын тулд дэвшилтэт функцийн туг удирдлагыг хэрэгжүүлэх шаардлагатай хэвээр байна.

Үүнийг хэрэгжүүлэхдээ та шинэ серверийн API дуудлагуудыг нэмэх эсвэл одоо байгаа дуудлагуудыг сайжруулах боломжтой. Хэрэв та үүнийг хийвэл шинэ функцээ шалгахын тулд Mocha тестийг (`test/serverApiTest.js`) шинэчлэх шаардлагатай. Хэрэв та шинэ API-үүдийг нэмбэл, бусад тестүүдэд ашигласан загварын дагуу шинэ `describe()` блокт оруулна уу. Бүртгүүлэхээсээ өмнө өгсөн тестүүд тэнцсэн хэвээр байгаа эсэхийг шалгаарай. Та Mongoose схемд шинэ шинж чанарууд нэмэх ёсгүй, гэхдээ та үүнийг илүү том өгөгдлийн багц дээр ажиллуулахын тулд шаардлагатай индексүүдийг нэмж болно.

Хичээлийн ажилтнууд үнэлгээ өгөхийн тулд таны аппликейшн дээр (хэрэв байгаа бол) Нарийвчилсан функцүүдийн тохиргоог идэвхжүүлж, нэмэлт кредит хэсэгт үнэлгээ өгөх эсэхийг тодорхойлохын тулд UI тоолох бөмбөлөгүүдийг хайж олох болно.

## Загварын оноо (5 оноо)

Хэрэв таны даалгаврт MVC бүтцийг зөв зохион байгуулж хэрэгжүүлсэн бол энэ оноог олгоно. Нэмж дурдахад, таны код болон модулиуд нь цэвэр, уншигдахуйц байх ёстой бөгөөд таны апп нь гадаад төрх, харагдацын хувьд "боломжийн сайхан" байх ёстой.

Илгээхээсээ өмнө `npm run lint` ажиллуулахаа мартуузай. Линтер нь алдаа гаргахгүй байх ёстой.

## Хүлээлгэн өгөх үр дүн

`project6` төсөлийг илгээхдээ стандарт класс `submission` `mechanism` ашиглана уу. Таны код ямар ч lint болон анхааруулгагүй байх ба өгсөн асуудлын шаардлагуудаа хангасан байх хэрэгтэй.

# Төсөл 7: Сесшн ба оролт

Энэ төсөлд хэрэглэгчдэд нэвтрэх, зураг дээр сэтгэгдэл бичих, шинэ зураг серверт хуулах замаар Төсөл 6-аар хийсэн ажлаа өргөтгөнө. Эдгээр шинэ функцийн нэмэлтүүд нь бүтэн стек (full stack) хөгжүүлэлт тул та фронтэнд (React программ) болон бакэнд (Node вэб сервер болон MongoDB өгөгдлийн сан) хоёуланг нь өөрчлөх шаардлагатай.

## Тохиргоо

Та өөрийн системд MongoDB болон Node.js суулгасан байх ёстой. Хэрвээ байхгүй бол [суулгах заавар](#) дагаж суулгаарай.

Төсөл 6-ийн нэгэн адил өмнөх төслийн хавтсын (`project6`) файлуудыг `project7` нэртэй шинэ хавтасруу хуулж эхэлнэ (`.babelrc` болон `.eslintrc.json` зэрэг зэрэг далдалсан бүх файлуудыг хуулсан эсэхийг шалгаарай). Энэ `zip` файлыг `project7` хавтас дотор задлана. Энэ шахсан файл хэд хэдэн шинэ файл нэмэх юм. `project7` хавтас дотор энэ төслийн бүх ажлаа хийгээрэй.

Энэ даалгаврыг хийхийн тулд та хэдэн шинэ `node_modules` татаж тэдгээрийг вэб сервер болон React программдаа нэмэх хэрэгтэй болно. Модуль бүрийн хувьд дараах командыг ашиглаж татаж авна уу:

```
npm install --save <module>
```

Энэ нь `<module>` нэртэй модулийг таны `node_modules` хавтаст татаж, `package.json` файлд модулийг (dependencies) нэмнэ. Модулийг вэб сервертээ нэмэхийн тулд `require` функцийг ашиглана. Та node модулиуддаа дараах Express дундын программ (middleware) татах шаардлагатай болно:

- `express-session` - Express сесшн нь сесшнийг өөрөө удирдах гэхээс илүүтэйгээр танд зориулсан Express дундын давхарга юм. Үүнийг лекцээр тайлбарласан.

- `body-parser` - `body-parser` нь HTTP хүсэлтийн бие хэсгийг (body) задлахад зориулсан Express дундын давхарга юм. Та үүнийг өөрсдийн серверийн API-д ашигладаг JSON форматаар кодлогдсон POST хүсэлтүүдийн бие хэсгийг задлахад ашиглаж болно. Жишээлбэл, `parameter_name` шинжтэй JSON объектыг хүсэлтийн бие хэсэгт дамжуулбал үүнд Express хүсэлт баригчаар `request.body.parameter_name` гэж хандах боломжтой.
- `multer` - `multer` нь зураг серверт хуулахад шаардлагатай олон хэсэгт маягтуудыг боловсруулах чадвартай, хүсэлтийг бие хэсгийг задлах өөр нэг Express дундын программ юм.

Ихэнх системд дараах командаар дээрх модулиудыг татдаг:

```
npm install --save express-session
npm install --save body-parser
npm install --save multer
```

`require` функцийг дуудах дараах мөрийг `webServer.js` файлд нэмнэ:

```
const session = require("express-session");
const bodyParser = require("body-parser");
const multer = require("multer");
```

Эцэст нь та дараах байдлаар Express `use` ашиглан `express-session` болон `body-parser` дундын программыг нэмэх хэрэгтэй:

```
app.use(session({secret: "secretKey", resave: false, saveU
ninitialized: false}));
app.use(bodyParser.json());
```

Энд `"secretKey"` нь сесшн күүкийг криптографийн аргаар хамгаалахад ашиглах нууц үг. Бид зураг серверт хуулахад `multer` дунд программыг ашиглах болно.

Хэрэв та 6-р төслийн 2-р даалгаврыг гүйцэтгэсэн бол загваруудыг татахдаа `axios` модулийг ашиглах хэрэгтэй. Хэрэв та Төсөл 6-д

хэрэгжүүлээгүй бол илүү дэлгэрэнгүйг `axios` баримт бичгээс харна уу.

Өмнөх даалгавартай адил бид серверийн API-г шалгах Mocha тестийг өгсөн тул фронт-энд хэсгээр хөгжүүлэхээс өмнө бак-энд код найдвартай ажиллаж буйд итгэлтэй байх боломжийг олгоно. Энэ боломжийн хариуд та бидний тодорхойлсон серверийн API-г хэрэгжүүлэх хүрээнд хязгаарлагдаж хөгжүүлнэ. Энэ нь та кодоо илгээхээсээ өмнө бидний өгсөн тестийг давж байгаа эсэхээ шалгах боломжтой гэсэн үг. Дэлгэрэнгүй мэдээллийг доорх Тест хэсгээс үзнэ үү. Хэрэв та 6-р төслийн нэмэлт онооны даалгаврыг хийж API-аа өргөтгөсөн бол үүнийг шалгах тестээ өөрөө нэмж бичих шаардлагатай. Хэрэв энэ төсөлд нэмэлт онооны даалгаврыг хийхээр шийдсэн бол нэмэлт тестийг бас давах шаардлагатай болно.

Өмнөх төслийн нэгэн адил та MongoDB өгөгдлийн сангийн системээ асаах хэрэгтэй болно. MongoDB-г дараах командаар ажиллуулж эхлүүлнэ үү:

```
mongod (өгөгдлийн санг хаана байршуулснаас хамаарсан тодорхой аргументууд)
```

мөн дараах командыг ажиллуулж зургийн программын өгөгдлийг өгөгдлийн санд хуулна уу:

```
node loadDatabase.js
```

Энэ командын Төсөл 7-д өгсөн хувилбар нь хэрэглэгчийн объектыг `login_name` (овгийн `last_name` жижиг үсгээр бичсэн хувилбар) болон `"weak"` гэсэн нууц үгээр хуулдаг болохыг анхаарна уу. Эдгээр шинж нь зөвхөн моделийн схемд талбаруудыг нэмсний (1-р даалгавар) дараа `loadDatabase.js` кодыг ажиллуулах үед гарч ирнэ.

## Node.js вэб серверийг эхлүүлэх нь

Өгөгдлийн сангаа байгуулж системээ асаасны дараа вэб серверээ асаах хэрэгтэй болно. Хэдийгээр үүнийг өмнөх даалгавруудтай ижил командаар (жишээ нь, `node webServer.js`), хийж болох ч `webServer.js` кодыг өөрчлөх үед автоматаар дахин эхлүүлэх

программ ашиглан вэб серверийг ажиллуулах нь илүү тохиромжтой. Үгүй бол та хийсэн кодын өөрчлөлт бүрийн дараа вэб серверээ дахин асаахаа мартсан үед таны өөрчлөлт яагаад ажиллахгүй болсныг гайхан цаг хугацаа зарцуулж магадгүй.

Дараах команд

```
npm install -g nodemon
```

серверийг автоматаар дахин ажиллуулдаг **nodemon** нэртэй программыг суулгана (та **sudo** командыг ашиглах хэрэгтэй байж магадгүй). **-g** параметртэйгээр ажиллуулбал нь программыг global байдлаар суулгадаг тул та хаана ч ажиллуулах боломжтой. **project7** хавтас дотроос дараах командаар вэб серверээ асаана:

```
nodemon webServer.js
```

Энэ команд вэб серверийг ажиллуулж, **webServer.js** файл өөрчлөгдөх бүрт дахин эхлүүлэх болно. Хэрэв Node программаас гарахад хүргэдэг алдаа (жишээ нь, ЖаваСкрипт бичлэгийн дүрмийн алдаа) гарвал nodemon **webServer.js** файлыг өөрчлөх хүртэл хүлээж, дараа нь дахин эхлүүлэхийг оролдоно. Хэрэв та файлыг өөрчлөхгүйгээр вэб серверийг дахин эхлүүлэхийг хүсвэл nodemon командын **rs** гэсэн хоёр тэмдэгт командыг бичиж гараар дахин эхлүүлж болно. Та Node.js кодоос лог зурвасыг (logging messages) гаргаж харахдаа үүнийг өөрийн хөгжүүлэлтийн орчны цонх эсвэл терминал дээрээ ажиллуулах хэрэгтэй.

Төсөл 7-ийн шинэ файлуудаар өөрийн Photo Share программаа шинэчилж өгөгдлийн сан болон вэб серверийг ажиллуулсны дараа, даалгавараа хийж эхлэхээсээ өмнө программаа ажиллаж байгаа эсэхийг шалгана.

## Даалгавар 1: Энгийн нэвтрэлт (15 оноо)

Хэрэглэгч нэвтэрдэг гэсэн ойлголттой болохын тулд зургийн аппаа өргөтгөнө. Хэрэв хэрэглэгч нэвтэрсэн бол хэрэгслийн талбарт (toolbar) "Hi <firstname>" гэсэн товч зурвас байх бөгөөд энд

<firstname> нь нэвтэрсэн хэрэглэгчийн нэр байна. Хэрэгслийн талбар нь хэрэглэгчийг гаргах "Logout" товчийг мөн агуулах ёстой.

Хэрэв нэвтэрсэн хэрэглэгч байхгүй бол хэрэгслийн талбарт "Please Login" гарч ирэх бөгөөд таны программын үндсэн хэсэгт `LoginRegister` нэртэй шинэ харагдац (view) компонент гарч ирнэ. `LoginRegister` компонент нь хэрэглэгчийг нэвтрэх арга замыг, доорх 4-р даалгаврын нэг хэсэг болох шинэ хэрэглэгчийг бүртгэх боломжийг хамт зааж өгнө. Өөр харагдац руу шилжих бусад бүх оролдлого (жишээ нь, гүн холбоос) нь хэрэглэгч нэвтрэх ороогүй тохиолдолд `LoginRegister` харагдац руу шилжүүлдэг байна. (Хэрэв та үүнийг хэрхэн хэрэгжүүлэхээ мэдэхгүй байгаа бол зөвлөмж хэсгийг үзнэ үү.) Үүнээс гадна, хэрэв хэрэглэгч нэвтрээгүй бол зүүн талд байгаа хэрэглэгчийн жагсаалтыг харуулах ёсгүй. (401 (Unauthorized) статусыг буцаахын тулд серверийн эндпойнтуудыг өөрчлөх талаар энэ даалгаврын хамгийн доод тайлбарыг үзнэ үү).

Хэрэглэгч амжилттай нэвтэрсэн үед харагдах байдал нь хэрэглэгчийн дэлгэрэнгүй мэдээллийг харуулах хэсэг рүү шилжих ёстой. Хэрэв хэрэглэгчийн нэвтрэлт амжилтгүй болвол (жишээ нь, `login_name`-тэй хэрэглэгч байхгүй) харагдац нь зохих алдааны мэдэгдлийг мэдээлэх бөгөөд хэрэглэгч дахин оролдох боломжийг олгох юм.

Нэвтэрсэн хэрэглэгч гэх ойлголтыг зургийн аппдаа оруулахын тулд бакэндийн хэрэгжүүлэлтийг өргөтгөнө. Энэ өөрчлөлтийг хийхдээ та өгөгдлийн сангийн схем болон вэб серверийн API хоёуланг нь өөрчлөх шаардлагатай.

`User` дээр шинэ хэсэг (property) `login_name` нэмэхийн тулд Mongoose схемийг өргөтгөнө үү. Энэ хэсэг нь хэрэглэгчийн нэвтрэх үед бичих үед танигчийг (тэдний "нэвтрэх нэр") агуулсан тэмдэгт мөр юм.

Вэб серверийн API-г өөрчлөн хэрэглэгчийг нэвтрэх, гаргахын тулд 2 шинэ REST API дуудлагыг хийх хэрэгтэй. Өмнөх даалгаврын нэгэн адил бид загварын өгөгдлийг дамжуулахын тулд JSON кодлогдсон биетэй HTTP хүсэлтийг ашиглах болно. API нь POST хүсэлтийг дараах зорилгоор ашигладаг:

- `/admin/login` - Зургийн программын `LoginRegister` харагдац нь хэрэглэгч рүү нэвтрэх боломжийг олгодог. POST хүсэлттэй

JSON-кодлогдсон хэсэг нь property `login_name` (одоохондоо нууц үг байхгүй) агуулсан байх ёстой бөгөөд таны апп-д нэвтэрсэн хэрэглэгчдэд шаардлагатай мэдээллийг агуулсан байх ёстой. Нэвтрэх амжилтгүй болсон тохиолдолд (жишээ нь , `login_name` хүчинтэй бүртгэл биш) 400 (муу хүсэлт) HTTP статусыг буцаана. Хүсэлтийн бие дэх параметрт `request.body.parameter_name` ашиглан ханддаг. Нэвтрэх бүртгэлийн зохицуулагч нь өгөгдсөн `login_name`-тэй хэрэглэгч байгаа эсэхийг шалгах ёстой гэдгийг анхаарна уу. Хэрэв тийм бол энэ нь хэрэглэгч нэвтэрсэн эсэхийг мэдэх шаардлагатай бусад хүсэлт боловсруулагчаар шалгах боломжтой зарим мэдээллийг Экспресс сессэд хадгалдаг. `POST /admin/login` -ийн буцаах хэсэг нь таны нэвтэрсэн хэрэглэгчийн шинж чанар буюу вэб программын хэрэгцээ байх ёстой. Манай шалгах хэсгүүдэд `_id` property шаарддаг боловч та бусад хэрэглэгчийн объектын шинж чанаруудыг оруулж болно. 4-р асуудалд нэмсэн зүйлээс болж хэрэглэгчийн объектыг бүхэлд нь буцаах нь дэмий болдог.

- `/admin/logout` -Энэ URL руу хоосон биетэй POST хүсэлт нь сессэд (session) хадгалагдсан мэдээллийг устгаснаар хэрэглэгч оролтоос гарах болно. Хэрэглэгч одоогоор нэвтрээгүй байгаа үед HTTP статус 400 (муу хүсэлт) буцаж ирэх ёстой.

Нэвтрэх/гаралтыг зохицуулахын тулд вэб серверийг шинэчлэх нэг хэсэг болгон хэрэв сессийн төлөв нь хэрэглэгч нэвтрээгүй гэвэл чих бүх хүсэлтүүдээ (`/admin/login` болон `/admin/logout`-аас бусад) 401 (Unauthorized) статусыг арилгахын тулд дахин шинэчлэх хэрэгтэй.

## Даалгавар 2: Шинэ сэтгэгдэл (15 оноо)

Хэрэглэгчийн нэвтрэлтийг хэрэгжүүлсний дараа дараагийн алхам бол зураг дээр сэтгэгдэл нэмэх боломжийг хэрэгжүүлэх явдал юм. Зургийн тайлбарыг харуулах зургийн дэлгэрэнгүй харагдац дээр одоо нэвтэрсэн хэрэглэгчид зурган дээрээ сэтгэгдэл нэмэх боломжийг нэмнэ үү. Та энэ функцийн хэрэглэгчийн интерфэйсийг (жишээ нь, поп-ап харилцах цонх, оруулах талбар гэх мэт) зохион бүтээх

хэрэгтэй. Үүнийг хэрхэн ашиглах, ямар зургийн тухай сэтгэгдэл бичих нь тодорхой байх ёстой. Зургийн дэлгэц болон түүний тайлбарыг шинээр нэмсэн сэтгэгдлийг тусгахын тулд нэн даруй шинэчлэх шаардлагатай.

Арын талын дэмжлэгийн хувьд вэб серверийн API-г дараах HTTP POST API-аар өргөтгөнө үү:

- `/commentsOfPhoto/:photo_id` - ID нь `photo_id` гэсэн зурган дээр сэтгэгдэл нэмнэ. POST хүсэлтийн үндсэн хэсэг нь тайлбарын текстийг агуулсан нэг property тайлбар бүхий `comment`-ийг JSON кодлогдсон бие байлгах ёстой. Зураг дээр үүсгэсэн сэтгэгдэл бичих объект нь нэвтэрсэн хэрэглэгчийн таниулбар болон сэтгэгдэл үүсгэсэн цагийг агуулсан байх ёстой. Таны хэрэгжилт 400 статустай (Bad request) хоосон сэтгэгдэл бичих боломжгүй байлгах ёстой.

## Даалгавар 3: Зураг байршуулах (15 оноо)

Хэрэглэгчдэд шинэ зураг нэмэх боломжийг бий болгоно. Хэрэглэгч нэвтэрсэн үед үндсэн хэрэгслийн (toolbar) мөрөнд "Зураг нэмэх" гэсэн товчлуур байх ёстой бөгөөд энэ нь одоогийн нэвтэрсэн хэрэглэгчийг апп руу зураг оруулах боломжийг олгодог. Зургийг амжилттай байршуулсны дараа энэ нь хэрэглэгчийн зургийн хуудсан дээр автоматаар гарч ирэх эсвэл хамгийн багадаа зургийг нэмсэн гэсэн тэмдэгтэй байх ёстой. Бид танд HTML ашиглан файл байршуулах жишээг доорх зөвлөмж(hint) хэсэгт өгөх болно.

URL дэх POST хүсэлтийг дэмжихийн тулд вэб серверийг өргөтгөх нь:

- `/photos/new` - Одоо байгаа хэрэглэгчийн зургийг байршуулна уу. POST хүсэлтийн үндсэн хэсэг нь файл байх ёстой (доорх зөвлөмжийг (hints) үзнэ үү). Байршуулсан файлуудыг `images` ргийн санд өөрийн үүсгэсэн өвөрмөц нэрээр байршуулах ёстой. Өвөрмөц нэрийг үүсгэсэн өгөгдөл болон нэвтэрсэн хэрэглэгчийн ID-ийн хамт таны шинээр үүсгэсэн Фото объектод оруулах ёстой. Хэрэв POST хүсэлтэд файл байхгүй бол 400 гэсэн хариултын статусыг буцаана. Энэ

талаар тусламж авахыг хүсвэл зөвлөмж (hints) хэсгээс үзнэ үү.

## Даалгавар 4: Бүртгэл ба нууц үг (15 оноо)

Шинэ хэрэглэгчийн бүртгэл болон нууц үгийг дэмжихийн тулд `LoginRegister` харах бүрэлдэхүүн (view component) хэсгийг сайжруулна уу. Шинэ хэрэглэгчийн бүртгэл болон нууц үгийг дэмжихийн тулд `LoginRegister` харах бүрэлдэхүүн (view component) хэсгийг сайжруулна уу. Нууц үг оруулах хэсэг нэмэхийн тулд нэвтрэх хэсгийг сунгана уу. Хэрэглэгчийн объектын бүх хэсгийг дүүргэх боломжийг олгодог бүртгэлийн хэсгийг нэмнэ үү. Хэрэглэгч нууц үгээ бичдэг гэсэн магадлалыг багасгахын тулд харагдац нь нууц үгийн хэсгийг дахин хуулбарлан агуулсан байх ёстой бөгөөд харагдац нь зөвхөн хэрэглэгчийн оруулсан нууц үгийн болон хуулбар хоёр талбар ижил байвал үүсгэнэ. Аюулгүй байдлын практик нь хэрэглэгчийн оруулсан нууц үг нь харагдах байдалд харагдахгүй байхыг шаарддаг. Бүртгэлийг хуудасны доод хэсэгт байрлах "Register Me" гэсэн товчлуураар идэвхжүүлнэ. Товчлуур дээр дарахад **specifically** яагаад ажиллахгүй байгааг тайлбарласан алдааг харуулах эсвэл амжилттай болсон тухай мэдээлэл ба бүртгэлийн маягтын оруулах талбаруудыг арилгах шаардлагатай.

Backend-ийн хувьд `User` ийн объектын схемийг нууц үг хадгалах `password` бүхий мөрийн талбарыг өргөтгө. Энэ бол үнэхээр найдваргүй зүйл. Хэрэв та маш их найдваргүй зүйлийг хэрэгжүүлмээргүй бол доорх Нэмэлт кредитийг харна уу.

URL руу POST хүсэлтийг дэмжихийн тулд вэб серверийг өргөтгөх нь :

- `/user` - Хэрэглэгч бүртгүүлэх боломжийг олгоно. Бүртгэлийн POST нь дараах шинж чанаруудтай JSON кодлогдсон биеийг авдаг: (`login_name`, `password`, `first_name`, `last_name`, `location`, `description`, `occupation`). Нийтлэлийн хүсэлтийг зохицуулагч нь шинэ `login_name` зааж өгсөн бөгөөд аль хэдийн байхгүй эсэхийг шалгах ёстой. `first_name`, `last_name`, болон `password` нь хоосон биш мөр байх ёстой (бусад талбарууд хоосон байж болно). Хэрэв мэдээлэл хүчинтэй бол

өгөгдлийн санд шинэ хэрэглэгч бий болно. Хариултын хэсэг нь таны вэб программд хэрэгтэй шинж чанарууд байх ёстой. Манай тестүүд `login_name` шинж чанарыг агуулсан байхыг шаарддаг. Хэрэв алдаа гарсан бол хариу нь 400 статус болон алдааг харуулсан мөрийг буцаана.

Нууц үгээр нэвтрэхийг дэмжихийн тулд `LoginRegister` харагдацыг сайжруулж, `/admin/login` руу илгээх хүсэлтийн нэг хэсэг болгон шалгана уу.

## Нэмэлт оноо #1: Давсалсан нууц үг (5 оноо)

Давслах аргыг *salting* хэрэгжүүлснээр нууц үгийнхээ механизмын аюулгүй байдлыг сайжруулаарай. The salting mechanism is described in the next few paragraphs. Давслах механизмыг дараагийн хэдэн догол мөрөнд тайлбарласан болно. 4-р даалгаварт бидний хэрэгжүүлсэн тодорхой текст нууц үгийн механизмтай холбоотой даалгавар нь хэн нэгэн өгөгдлийн санд уншиж чаддаг бол (жишээлбэл, системийн администратор гэх мэт) бүх хэрэглэгчийн нууц үгийг хялбархан олж авах боломжтой.

Нууц үг болгонд SHA-1 гэх мэт мессежийг нэгтгэх функцийг хэрэглэж, өгөгдлийн санд зөвхөн мессежийн нэгтгэлийг хадгалах нь илүү сайн арга юм. SHA-1 нь нууц үг гэх мэт мөрийг оролт болгон авч, 40 тэмдэгтээс бүрдэх зургаан өнцөгт цифрүүдийн мөрийг мессежийн дайжест (*message digest* гэж нэрлэдэг) гаралт болгон гаргадаг. Гаралт нь хоёр property-той байна: нэгдүгээрт, цифр нь оролтын мөрөнд өвөрмөц гарын үсгийг өгдөг (ижил дайжест-тэй хоёр өөр мөр үүсгэх тодорхой арга байхгүй); Хоёрдугаарт, мессежийн дайжест-ийг өгвөл уг дайжест-ийг үүсгэх мөрийг үүсгэх ямар ч мэдэгдэж байгаа арга байхгүй. Хэрэглэгч нууц үгээ тохируулах үед та Node `crypto` багцын `createHash` функцийг ажиллуулж тухайн нууц үгтэй тохирох SHA-1 дайжест үүсгэх ба өгөгдлийн санд зөвхөн дайжест хадгалах ёстой. Үүнийг хийсний дараа та нууц үгээ ашиглахгүй байж болно. Хэрэглэгч нэвтрэхийн тулд нууц үгээ оруулах үед дайжест тооцоолохын тулд `createHash` функцийг ажиллуулж, мэдээллийн санд хадгалагдаж буй мэдээлэлтэй

харьцуулна уу. Энэ аргын тусламжтайгаар та хэрэглэгч нэвтэрч орохдоо нууц үгээ зөв оруулсан эсэхийг шалгах боломжтой боловч хэрэв хэн нэгэн өгөгдлийн сангаас мэдээлэл уншвал тэр мэдээллийг ашиглах боломжгүй.

Гэсэн хэдий ч өмнөх догол мөрийн арга барил нь нэг дутагдалтай хэвээр байх болно. Халдагчид дайжест агуулсан өгөгдлийн сангийн хуулбарыг авсан гэж бодъё. SHA-1 функц нь олон нийтэд нээлттэй байдаг тул тэд энгийн нууц үгийг таахад хурдан толь бичгийн халдлага (*fast dictionary attack*) ашиглаж болно. Үүнийг хийхийн тулд халдагч үг бүрийг толь бичгээс авч, SHA-1-ийг ашиглан дайжест-ийг тооцоолно. Дараа нь халдагчид өгөгдлийн сан дахь дайжест бүрийг толь бичигт байгаа дижестүүдтэй харьцуулан шалгадаг (үүнийг бүх толь бичгийг хэш хүснэгтэд оруулснаар маш хурдан хийж болно). Хэрэв хэрэглэгч нууц үгээ энгийн толь бичгийн үгийг сонгосон бол халдагч үүнийг хурдан таах боломжтой.

Толь бичгийн довтолгоог илүү төвөгтэй болгохын тулд та нууц үг давслах аргыг ашиглах ёстой. Хэрэглэгч нууц үгээ тохируулах үед SHA-1 дижестийг тооцоолохын өмнө санамсаргүй тоог тооцоолж, нууц үгтэй холбох ба (8 урттай крипто багцын `randomBytes` функц нь тохирох санамсаргүй тоог буцаана. Санамсаргүй тоог давс(*salt*) гэнэ. Дараа нь өгөгдлийн санд давс болон дайджестийг хоёуланг нь хадгална уу. Нэвтрэх үед нууц үгээ шалгахдаа өгөгдлийн сангаас давсыг авч, хэрэглэгчийн бичсэн нууц үгтэй холбож, өгөгдлийн сан дахь дайжесттэй харьцуулах зорилгоор энэ мөрийн хураангуйг тооцоолно. Энэ байдлыг ашиглан нэвтрэх өгөгдлийн санд хандсан халдагч нь дээр дурдсан толь бичгийн довтолгоог ашиглах боломжгүй бөгөөд энэ нь халдагч бүх мэдээллийг дахин тооцоолох шаардлагатай болно. Энэ нь толь бичгийн довтолгоог илүү үнэтэй болгодог.

Үүнийг хэрэгжүүлэхийн тулд хэрэглэгчийн схемээс нууц үгийн `property`-ийг устгаад нууц үгийн шинж чанарыг шинэ хоёр мөрийн шинж чанараар солино: `password_digest` and `salt`. Энэ механизмыг ашиглахын тулд хэрэглэгчийн бүртгэлийг шинэчилж, нэвтэрнэ үү.

Та `ICSI301password.js` нэртэй `node` модулийн файл үүсгэж, хэрэгжүүлэлтийг бичих хэрэгтэй. Энэ модуль нь дараах хоёр функцийг экспортлох ёстой:

```
/**
 * Return a salted and hashed password entry from a clear text password.
 * @param {string} clearTextPassword
 * @return {object} passwordEntry where passwordEntry is an
 * object with two
 * string properties:
 *   salt - The salt used for the password.
 *   hash - The sha1 hash of the password and salt.
 */
function makePasswordEntry(clearTextPassword) {
```

болон

```
/**
 * Return true if the specified clear text password and salt
 * generates the
 * specified hash.
 * @param {string} hash
 * @param {string} salt
 * @param {string} clearTextPassword
 * @return {boolean}
 */
function doesPasswordMatch(hash, salt, clearTextPassword) {
```

Бид энэ интерфэйсийг шалгадаг Mocha тест `test/passwordApiTest.js` файлыг өгсөн. Төслөө илгээхээсээ өмнө энэ тестийг давсан эсэхээ шалгана уу. Та `package.json` файлдаа `mocha` `serverApiTest.js` `sessionInputApiTest.js`

`passwordApiTest.js` скриптүүдийг нэмж өөрчлөөд `npm test` гэж эдгээр тестийг ажиллуулж болно. Энэ өөрчлөлтийг хийх нь та энэ нэмэлт оноог авна гэсэн үг юм. Мөн `loadDatabase.js` скриптэд `ICSI301password.js` файлыг дуудаж өгсөнөөр шинэ хэрэглэгчийн объектуудад шаардлагатай нууц үгийн шинжүүдийг зөв үүсгэхэд ашиглана.

## Нэмэлт оноо #2: Хөтөчийг шинэчлэх (5 оноо)

Энэхүү даалгаврын энгийн хэсгүүдэд хялбар болгох үүднээс бид танд программын сесшний төлөвийг ЖаваСкриптийн санах ойд хадгалахыг зөвшөөрч байна. Хэдийгээр энэ нь хэрэгжилтийг хөнгөвчлөх боловч хөтчийг шинэчлэх нь программыг хэн нэвтэрсэнийг мартахад хүргэдэг.

Нэвтрэх сесшний дэмжлэгийг нэмэхээс өмнөх шиг хөтчийн шинэчлэлтийг зохицуулахын тулд программаа өргөтгө. Таны схем нь хэрэглэгчдэд хөтчийг шинэчлэх боловч нэвтэрсэн хэвээр байх боломжийг олгох ёстой. Та хүссэн хэрэгжүүлэх арга техникээ чөлөөтэй ашиглаж болно, гэхдээ тэдгээр нь:

- Энэ даалгаврын бусад хэсгүүд, түүний дотор Mocha тестүүдтэй хоцрогдсон нийцтэй байдлыг хадгалах.
- Программын аюулгүй байдлыг алдагдуулахгүй байх.
- Ангийн даалгавар илгээх механизмыг ашиглан ирүүлсэн тохиолдолд ажиллана.

## Загварын оноо (5 оноо)

Хэрэв даалгаврын шийдэл нь MVC зохион байгуулалттай, программ хангамжийн стекийн дүрмийг дагаж мөрддөг, ESLINT-ийн анхааруулгагүй ЖаваСкрипт код байвал эдгээр оноог олгоно. Нэмж дурдахад, таны код, загварууд нь цэвэрхэн, уншихад хялбар байх ёстой бөгөөд таны вэб хуудаснууд гадаад төрх, тав тухтай байдлын хувьд дор хаяж "боломжийн сайхан" байх ёстой.

# Тест

Тестийн хувьд бид:

- Шинэ API дуудлагын зарим тестийн хамрах хүрээг хангадаг `test/sessionInputApiTest.js` нэртэй шинэ тест файлыг өгөх.
- Одоо байгаа `test/serverApiTest.js` шинэчилж хэрэглэгч рүү нэвтэрч, API дуудлага бүр дээр сессийн мэдээллийг өгнө.

Бидний шинэчлэлт нь `serverApiTest.js` файлыг дарж бичих тул нэмэлт зээлийн нэг хэсэг болгон Төсөл 6-д нэмсэн аливаа өөрчлөлтийг буцааж мэдээлэх шаардлагатай болно. Нөөц дамжуулахад туслахын тулд (эсвэл та зүгээр л сонирхож байгаа бол) бид HTTP хүсэлтийн дуудлага бүрийн хүсэлтийн сонголтуудад күүки толгойн түлхүүр-утгийн хосыг нэмсэн гэдгийг анхаарна уу. Дэлгэрэнгүйг тестийн файлаас үзнэ үү.

**Төсөл 6-д байгаа шиг, шалгалт өгөхөөсөө өмнө өгсөн шалгалтуудыг заавал давах хэрэгтэй, учир нь таны үнэлгээний тодорхой хэсэг нь эдгээр шалгалтын хэд нь тэнцсэнээс хамаарна.**

Туршилтууд нь өгөгдлийн санд зөвхөн `loadDatabase.js` бъектуудыг агуулж байгааг анхаарна уу. Та тестийг эхлүүлэхийн өмнө `loadDatabase.js` ажиллуулах хэрэгтэй.

Ямар нэгэн тест хийхээсээ өмнө `project7/test` directory-д `npm install` командыг бичиж `test/node_modules` шинэчлэгдсэн эсэхийг шалгах хэрэгтэй.

Үү Та 2 багцыг хуучирсан тухай анхааруулга авч магадгүй (жишээ нь `request` болон `har-validator`). Эдгээр нь бидний туршилтад ашигладаг номын сангийн аль нэг нь засвар үйлчилгээ хийж байгаатай холбоотой сэрэмжлүүлэг тул та эдгээрийг үл тоомсорлож болно. Та өмнөх шигээ `npm run test` командыг ашиглан тестүүдийг ажиллуулж болно.

## Зөвлөмж

Та одоо MongoDB өгөгдлийн санг шинэчлэх код бичиж байгаа тул таны кодын алдаа нь өгөгдлийн сангийн агуулгыг гэмтээж болзошгүй юм. `loadDatabase.js` программыг дахин ажиллуулснаар таны өгөгдлийн санг цэвэр төлөвт(clean state) дахин тохируулна. Туршилтууд таныг run бүрийн өмнө үүнийг хийсэн гэж үздэг.

## Даалгавар 1

Энэ асуудал нь таныг React программ болон вэб серверт нэвтэрсэн хэрэглэгчийн тухай ойлголтыг авахыг шаарддаг. Та өөрийн апп-аас вэб сервер рүү POST хүсэлт үүсгэхийн тулд загвараа буулгаж, сессийн төлөвийг тохируулах хэрэгтэй. Сервер талын функцэд та үүний эсрэг ажиллуулж болох тестүүд байдаг.

Хэрэв та өөрийн `LoginRegister` сбүрэлдэхүүн хэсгийг бичиж, зургийн программдаа нэмээд `/login-register` зам руу чиглүүлсэн гэж үзвэл бид хэрэглэгч нэвтэрч ороогүй тохиолдолд тухайн харагдацыг харуулах хэрэгтэй. React Router нь ашигтай үйлчилгээ үзүүлдэг. Дахин чиглүүлэлттэй ажиллах арга: [React Router Redirect](#). Хэрэв хэрэглэгч нэвтэрч ороогүй бол `/login-register` руу дахин чиглүүлэхийн тулд та маршрутаа өөрчилж болно. Жишээлбэл, хэрэв танд дараах маршрут байгаа бол:

```
<Route path="/users/:id" component={UserDetail} />
```

болон `this.userIsLoggedIn`, бүрэлдэхүүн хэсгийн өмчийг та дараах нөхцөлөөр багцалж `/login-register` руу шилжүүлж болно:

```
{
  this.userIsLoggedIn ?
    <Route path="/users/:id" component={UserDetail} />
    :
    <Redirect path="/users/:id" to="/login-register" />
}
```

Даалгавар 4 нь таны 1-р асуудалд зориулж бичсэн кодыг сайжруулдаг тул та Асуудлын 1-ийн шийдлийг бүтээхээсээ өмнө түүний хэрэгцээг ойлгохыг хүсч болно.

## Даалгавар 2

Энэ асуудал нь 1-р асуудлыг шийдэхэд шаардлагатай ихэнх зүйлийг дахин ашиглах болно. Нэг ялгаа нь comment нь манай схемийн бие даасан объект биш юм. Тэдгээрийг фото объект дотор суулгасан тул та объект үүсгэхийн оронд объектын шинэчлэлт хийх шаардлагатай болно.

## Даалгавар 3

Хөтөчүүд JavaScript кодыг дурын файлуудыг уншихыг хүсэхгүй байгаа тул JavaScript фреймворкуудад файл байршуулах нь хэцүү байдаг. Хөтөчүүд нь хэрэглэгчдэд файлаа сонгоод агуулгыг нь вэб серверт илгээх боломжийг олгодог интерфэйсээр хангадаг. Бид энэ механизмыг ашиглан манай зургийн аппликейшнд зураг байршуулах боломжтой.

React нь HTML-ийн `input with type="file"` ашиглан хэрэглэгчдэд файл сонгох боломжийг бий болгоно. Дараах мөрийг таны зургийн аппликейшн дээр нэмсэн

```
<input type="file" accept="image/*" ref={(domFileRef)} => { this.uploadInput = domFileRef; }} />
```

нь хөтөчийг(Browser) "Choose File" гэсэн хэсэгтэй товчлуурыг нэмж оруулах бөгөөд хэрэглэгч дотоод файлаа сонгох боломжтой. Хэрэглэгч файлыг сонгох үед `this.uploadInput` нь DOM файлыг авч вэб сервер рүү илгээх боломжтой оролтын элементээс `DOM FileList` агуулна. Та мөн доор тайлбарласны дагуу фото файлыг серверт илгээхийн тулд хэрэглэгч дарах товчлууртай байх шаардлагатай.

Сонгосон файлыг DOM-д оруулсны дараа бид үүнийг `DOM form` нэмж, вэб сервер рүү POST хүсэлтээр илгээх боломжтой. Дараах код нь үүнийг хийх жишээг өгдөг. Энэ нь оролтын элементийн DOM файлын жагсаалтыг `this.uploadInput`-д оруулсан гэж үздэг.

```
/**
 * Called when user presses the update button.
 */
```

```

handleUploadButtonClicked = (e) => {
  e.preventDefault();
  if (this.uploadInput.files.length > 0) {
    // Create a DOM form and add the file to it under the name
    // uploadedphoto
    const domForm = new FormData();
    domForm.append('uploadedphoto', this.uploadInput.files[0]);
    axios.post('/photos/new', domForm)
      .then((res) => {
        console.log(res);
      })
      .catch(err => console.log(`POST ERR: ${err}`));
  }
}

```

Body-parser ашиглан экспресс нь файл агуулсан маягттай POST хүсэлтийг зохицуулах боломжгүй боловч **multer** нэртэй дунд программаар хийх боломжтой. Мультиерийн require-ийн дараа дараах мөрийг оруулна уу:

```

const processFormBody = multer({storage: multer.memoryStorage()})
  .single('uploadedphoto');

```

**processFormBody** нь бидний **/photos/new** -д зориулсан шуудангийн хүсэлтийг зохицуулагчдаа ашиглаж болох функц юм. **processFormBody** нь "uploadedphoto" нэртэй талбарын маягтыг харж, тэндээс файлыг татаж аваад хүсэлтийн объект дээр **file** нэртэй өмчийн мэдээллийг байрлуулна. Дараах код нь таны шуудангийн хүсэлтийн зохицуулагч дээр үүнийг хэрхэн дуудах талаар санааг өгнө:

```

processFormBody(request, response, function (err) {
  if (err || !request.file) {

```

```

    // XXX - Insert error handling code here.
    return;
}

// request.file has the following properties of interest:
//   fieldname      - Should be 'uploadedphoto' since that is
//                   what we sent
//   originalname    - The name of the file the user uploaded
//   mimetype        - The mimetype of the image
//                   (e.g., 'image/jpeg',
//                   'image/png')
//   buffer          - A node Buffer containing the contents of
//                   the file
//   size            - The size of the file in bytes

// XXX - Do some validation here.

// We need to create the file in the directory "images"
// under an unique name.

// We make the original file name unique by adding a
// unique prefix with a
// timestamp.
const timestamp = new Date().valueOf();
const filename = 'U' + String(timestamp) +
                request.file.originalname;

fs.writeFile("./images/" + filename, request.file.buffer, func
tion (err) {
    // XXX - Once you have the file written into your images
    // directory under the

```

```
// name filename you can create the Photo object in
// the database

});

});
```

Дээрх кодын фрагмент нь Node.js багц `fs`-г ашигладаг тул та үүнийг `webServer.js`-дээ оруулах хэрэгтэй болно:

```
const fs = require("fs");
```

## Даалгавар 4

Энэ асуудал нь өмнөх асуудлуудад шаардлагатай арга техникийг нэгтгэсэн болно.

## Хүлээлгэн өгөх үр дүн

`project7` илгээхдээ стандарт `submission mechanism`-ийг ашиглана. Please clean up your project directory Илгээх зааварт заасны дагуу илгээхээсээ өмнө төслийн лавлахаа (directory) цэвэрлэнэ үү. Нэмж дурдахад, зургийн лавлахдаа байршуулсан бүх зургийг устгана уу. Хэрэв та "Зөвлөгөө (Hint)" хэсэгт байршуулсан зургуудыг нэрлэх дүрмийг баримталсан бол `rm -f images/U*` командыг ашиглан `project7`-аас устгаж болно.

# Төсөл 8: Фото зургийн программ спринт

[Программ хангамжийн шалмаг](#) (agile) хөгжүүлэх арга зүйг ашигладаг удирдлагын шинэ баг таны зургийн аппын хөгжүүлэлтийг удирдаж эхэллээ. Дараагийн [спринтийн](#) удирдлагын баг хувьд та төслийг хэрэгжүүлэх инженерийн хувьд сонгох боломжтой ажлуудыг багтаасан [хэрэглэгчийн түүхийн скрам самбарыг](#) (scrum board) тодорхойлсон. Түүх бүрт хэд хэдэн стори оноо оноодог бөгөөд спринтийн хувьд танд дор хаяж 12 оноотой бүрэн хэмжээний стори хэрэгтэй болно.

## Тохиргоо

Та өөрийн системд MongoDB болон Node.js суулгасан байх ёстой. Үгүй бол одоо суулгах зааврыг дагана уу.

7-р төслийн нэгэн адил та өмнөх төслийн хавтас ([project7](#))-ийн хуулбарыг [project8](#) нэртэй сан болгон хийж эхэлнэ. [project8](#) хавтас руу [энэ зип](#) файлын агуулгыг задлана. Энэхүү зип файл нь илгээх үйл явцын нэг хэсэг болгон бөглөх маягт агуулсан submission.txt нэртэй файлыг нэмэх болно. Энэ төслийн бүх ажлыг [project8](#) хавтас хэсэгт хийгээрэй.

7-р төслийн нэгэн адил танд MongoDB болон таны вэб сервер ажиллах шаардлагатай болно.

## Даалгавар 1: Зарим түүхийг хэрэгжүүлэх (75 оноо)

Төслийн скрам самбарыг доор харуулав. Хэрэглэгчийн түүх бүр түүхтэй холбогдож, уг функц юу хийх ёстойг зааж өгдөг. Та хийх

түүхээ сонгохдоо болгоомжтой байх хэрэгтэй. Түүхүүд хоорондоо харилцан уялдаатай байдаг бөгөөд тэдгээрийг хамтдаа хийхэд хялбар эсвэл хэцүү болгодог. Жишээлбэл, хэрэв та зураг дээр харагдах байдлын хяналт нэмэх тухай түүхийг хийхээр сонгосон бол үйл ажиллагааны мэдээллийн (feed) түүх нь хэрэглэгч харах боломжгүй зургийг буцааж өгөх ёсгүй байх ёстой.

Өмнөх даалгавруудаас ялгаатай нь та өгөгдлийн сангийн схем, вэб серверийн API болон ReactJS программын бүрэлдэхүүн хэсгүүдийг өөрчлөх зэрэг хүссэн өөрчлөлтийг хийх боломжтой. Танд хэрэгцээтэй гэж үзвэл гуравдагч талын программ хангамжийн багцыг ашиглаж болох ба илүүтэйгээр санал болгож байна.

Таны хөгжүүлэх эрх чөлөөг хязгаарлаж байгаа нэг зүйл бол таны зургийн аппликейшн класс-илгээх механизмаар дамжуулж, курсын ажилтнуудад үнэлгээ өгөхөд саадгүй ажиллах ёстой. Мөн түүхийг хэрэгжүүлэхдээ таны вэб программ ашиглалт/туршилтын явцад гацахгүй байхыг анхаарах хэрэгтэй (жишээ нь, webServer.js программ ажиллах явцад гацах ёсгүй). Дэлгэрэнгүй мэдээллийг дараагийн хэсгүүдээс илгээх мэдээллийг харна уу.

| Скрам самбар |                                                                        |            |
|--------------|------------------------------------------------------------------------|------------|
| #            | Хэрэглэгчийн түүх                                                      | Стори оноо |
| 1            | <a href="#">Хэрэглэгчийн профайлын дэлгэрэнгүй мэдээллийг өргөтгөх</a> | 3          |
| 2            | <a href="#">коммент дээр @mentions хийх</a>                            | 6          |
| 3            | <a href="#">Зургийн харагдах байдлыг хянах</a>                         | 4          |

|   |                                                                                           |    |
|---|-------------------------------------------------------------------------------------------|----|
| 4 | <a href="#">Activity feed</a>                                                             | 4  |
| 5 | <a href="#">Сэтгэгдэл, зураг, хэрэглэгчдийг устгах</a>                                    | 4  |
| 6 | <a href="#">Зурган дээр "like" санал өгөх</a>                                             | 4  |
| 7 | <a href="#">Хэрэглэгчдэд зориулсан дуртай зургийн жагсаалт</a>                            | 6  |
| 8 | <a href="#">Хажуугийн жагсаалт нь хэрэглэгчдийг шинэ үйл ажиллагаатай гэж тэмдэглэдэг</a> | 3  |
| 9 | <a href="#">Зурган дээр тагтлах</a>                                                       | 10 |

## Зөвлөмж

Загварын хувьд хэрэглэгчийн түүхүүд нь онцлог шинж чанаруудын хувьд нэг л утга санаатай үзүүлэлт биш юм. Энэ нь инженер та түүхэнд ашиглах арга барилыг заагаагүй газруудад боломжийн зүйлийг сонгох замаар дэлгэрэнгүй мэдээллийг бөглөх шаардлагатай гэсэн үг юм. Нэг үндсэн зарчим бол “Аппликешныг хэрэглэгч юу хийхийг хүсч байна вэ?” гэсэн асуултад хариулан хийх явдал юм.

## Дүгнэх

Даалгавраас бүтэн кредит(оноо) авахын тулд та хамгийн багадаа түүх хэсгээс 12 Стори оноо (story point) буюу (энэхүү даалгаврын 75 оноотой тэнцэх) хэрэглэгчийн түүхийг (story) хийх шаардлагатай. Таны 12-оос дээш оноо авсан Стори оноог даалгаврын оноотой адилаар тооцож хэрэглэнэ (жишээ нь, 12-оос илүү байх Стори оноог хөрвүүлэх байдал нь 1 Стори оноо = 1 даалгаврын оноо). Та нийтдээ

18 хүртэлх оноотой нэмэлт түүхийг хийх боломжтой. Өөрөөр хэлбэл та нийт 30 Стори оноо хийснээр 18 хүртэлх онооны нэмэлт кредит авах боломжтой. Бид үнэлгээний тооцоололдоо зөвхөн 5 түүхийг л оруулах болно.

Бид хэсэгчлэн хэрэгжүүлсэн түүхүүд нь хэсэгчлэн оноог нарийн байдлаар хуваарилахыг төлөвлөж байна. Цөөн түүхийг бүрэн гүйцэд хийх нь илүү олон түүхийг хэсэгчлэн хийхээс илүү сайн байх болно.

Хэрэв та npm-г ашиглан аливаа шинэ багцуудыг (package) татаж авах юм бол түүнийгээ package.json файлд нэмж оруулах хэрэгтэйг анхаарна уу (--save сонголтыг ашиглан үүнийг хийх хэрэгтэй) npm программыг ашиглахгүйгээр холбох бол өөрийн project8 хавтасдаа хадгалаарай, ингэснээр таны даалгаврыг үнэлэх харгалзагч багш үүнийг ашиглах боломжтой болно.

Энэ даалгаврыг үнэлэхэд та хэрэгжүүлсэн түүхүүдийнхээ талаар тайлбараа хавсаргах шаардлагатай. Project8 zip файл нь submission.txt файлыг татаж авсан. submission.txt нь бидэнд хэрэгтэй мэдээллийг цуглуулах маягтыг агуулсан. Үүнд таны Стори онооны тооцоолол, онооны хэмжээ, таны олж авсан нийт Стори оноо зэрэг багтана.

submission.txt файлаас гадна нэмэлтээр та өөрийн зургийн программ болон хэрэгжүүлсэн түүхийнхээ богино хэмжээний видеог оруулах ёстой. Видеоны урт нь  $12 \times k$  секундээс хэтрэхгүй байх ёстой бөгөөд k нь таны гүйцэтгэсэн Стори оноо юм. Ердийн видеоны урт нь 2 минут буюу түүнээс бага байна. Энэ нь аудио бүхий энгийн дэлгэцийн бичлэг байх ёстой. MacOS дээр та QuickTime, Windows дээр VLC ашиглан дэлгэцийн бичлэг хийх боломжтой. Видеог бүтээхэд хэт их цаг зарцуулах хэрэггүй. Та бичлэгийн чанараар үнэлэгдэхгүй гэдгийг анхаараарай.

Урт нь зөвшөөрөгдсөн дээд хэмжээтэй ойролцоо байгаа видеонууд нь бидний илгээх үйл явцад хэт том байх магадлалтай тул submission.txt

файлын холбоосыг бидэнд өгөх шаардлагатай байна.Видеоогоо YouTube эсвэл Vimeo гэх мэт видео хуваалцах боломжтой вэбсайт руу байршуулах нь иймэрхүү холбоосыг авахад хялбар арга болдог. Үнэлгээ тавих үйл явцын нэг хэсэг болгон бид ангийн хэлэлцүүлгийн форум дээр сайн хийгдсэн видеонуудаасаа хуваалцахаар төлөвлөж байна. Хэрвээ бичлэгээ бидэнтэй хуваалцахыг хүсвэл `submission.txt` файлд зааж өгөөрэй.

## Загварын оноо (5 оноо)

Хэрэв таны асуудлын шийдэл нь MVC-ийн зөв бүтэцтэй, MEAN стекийн жишгийг дагаж мөрддөг, ESLint анхааруулгагүй ЖаваСкрипт бол энэ оноог өгнө. Илгээхээсээ өмнө ``npm run lint`` командыг ажиллуулж программмыг алдааг шалгах хэрэгтэй. Нэмж дурдахад таны бичсэн код, загварууд цэвэр, уншихад хялбар байх ёстой бөгөөд таны вэб хуудсууд дор хаяж "боломжийн сайхан" харагдах байдал гоё байх ёстой.

Анхаар: `npm` сангаас татах боломжгүй модуль (`dependencies`), жишээ нь, `bower` сангаас авах модулийг суулгахдаа ``npm run lint`` скрипт команд нь таны модулийг агуулсан хавтасыг оруулахгүй байхаар `package.json` файлыг өөрчилнө үү. Ингэснээр ESLint команд нь зөвхөн таны өөрийн JS файлуудыг шалгах бөгөөд таны ашигладаг сан/модульд харгалзах JS файлуудыг алгасах юм. Энэ нь таны линтерийн шалгалтыг давахгүй болгож үнэлгээ өгөх боломжгүйд хүргэж болзошгүй.

## Илгээхдээ анхаарах зүйлс

Таны фото аппыг үнэлэхэд шаардлагатай бүх файлыг (зөвхөн `package.json` файлд заасан файлуудаас бусад нэмэлт файл шаардагдахгүйгээр) багцалж оруулах шаардлагатай. Хичээлийн багш нар таны бүх хавтсыг хуулж авна, гэхдээ `node_modules/` болон `compiled/` дэд хавтаснуудыг оруулахгүй. Эдгээр хавтасыг (ялангуяа

test/ хавтасны доторх node\_modules/ хавтсыг ч мөн адил) устгасан эсэхээ шалгаарай.

Дараах зүйлсийг шалгах хэрэгтэй:

- Төслийнхөө явцад ашигласан бүх модуль программыг package.json файлд заасан эсвэл төслийн хавтасны доторх дэд хавтас хэлбэрээр (гэхдээ node\_modules/ биш) байршуулсан байх ёстой.
  - npm install --save <module\_name> командыг ашигласнаар тухайн модулийг татаж авч package.json-д автоматаар нэмнэ.
  - Төслөө өгөхөөс өмнө node\_modules/ хавтсыг устгаад төслийн хавтас дотор npm install гэж бичиж ажиллуулж үзнэ үү. Дараа нь вэб апп хэвийн ажиллаж байгаа эсэхийг шалгаарай.
- loadDatabase.js скрипт файл байх ёстой бөгөөд MongoDB өгөгдлийн санд тохирох схем болон объектуудыг ачааллах боломжтой байх ёстой.
  - Шинэ схем нэмсэн бол уг скриптийг тохируулж өөрчлөх шаардлагатай байж магадгүй (жишээлбэл, шинэ цуглуулга (collection) оруулсан бол өгөгдлийг суулгахаас өмнө тухайн цуглуулгаас бүх объектуудыг устгах шаардлагатай).
  - mongod командыг дахин эхлүүлж, node loadDatabase.js командыг ажиллуулж, өгөгдлийн санг шинээр үүсгэж, суулгаж шалгах хэрэгтэй. Үүний дагуу таны апп ажиллахгүй бол бид шалгаж чадахгүй болно.
- node webServer.js команд вэб серверийг эхлүүлж, стандарт порт дээр localhost дахь MongoDB өгөгдлийн сантай холбогдож ажиллах ёстой. <http://localhost:3000/photo-share.html> хаягаар таны апп нээгдэх ёстой.

- Вэб сервер болон вэб аппаа сайтар шалгаж турших хэрэгтэй (жишээлбэл, хэрэглээний явцад сервер гацаж унахгүй байх ёстой).
- Вэб аппыг ажиллуулахад шаардлагатай бусад бүх файл таны оруулах сүүлийн хувилбарын төслийн хавтсан дотор орсон байх ёстой.
  - Үүнд өгөгдлийн санд холбоос нь орсон зургийн файлууд ч багтана.
- submission.txt файлыг бүрэн бөглөж төслийн хавтсанд хуулсан байх ёстой.
- Фото аппын танилцуулга богино видео бичлэг YouTube эсвэл Vimeo порталд байршуулсан байх ба түүний холбоос submission.txt файлд орсон байх ёстой.
- ESLint командыг ажиллуулахад таны ЖаваСкрипт код шалгуур хангаж, алдаагүй байх шаардлагатай. Дэлгэрэнгүйг зааврыг “Загвар” хэсгээс үзнэ үү.

Дээрх зааврын ихэнх нь 7-р төслийн даалгаварт ч мөн хамаарна. Хэрэв шинээр өөрчилсөн (жишээлбэл, шинэ программ импортлох, схемд талбар нэмэх гэх мэт) бол өөрчлөлтийг дахин шалгах шаардлагатай. Тодорхой онцгойлж заасан хэсгийг дагаж мөрдөх нь маш чухал юм. Дагаж мөрдөж чадаагүй бол таны вэб аппыг ажиллуулаад дүгнэж чадахгүй байж магадгүй.

1. ``mongod --dbpath /usr/local/var/mongodb``, (өгөгдлийн сангийн зам өөр байж болно),
2. ``npm install``,
3. ``node loadDatabase.js``, ба
4. ``node webServer.js``

(мөн зарим тохиолдолд таны submission.txt файлд тайлбарласан нэмэлт зааврыг дагана).

## Загвар (5 оноо)

Энэ оноог лекц дээр хэлэлцсэн MVC загварын зөвлөмжийг дагаж зөв зохион байгуулсан тохиолдолд өгнө. Таны код DOM орчинд шууд хандаж удирдсан байж болохгүй. Нэмж хэлэхэд, таны код болон загвар цэвэрхэн, уншууртай байх ёстой. Илгээхээсээ өмнө ESLint багажийг ажиллуулсан байх хэрэгтэй. ESLint ямар нэг алдаа өгөөгүй байна.

## Хүлээлгэн өгөх үр дүн

Стандарт классын даалгавар өгөх системийг ашиглан бүтэн аппликэйшнээ (project8 хавтас дахь бүх зүйл) оруулна уу. Даалгавраа өгөхөөс өмнө зааврын дагуу төслийн хавтсаа цэгцэлж цэвэрлэхээ мартуузай. Ялангуяа test/ хавтсан доторх node\_modules/ хавтсыг устгах хэрэгтэйг анхаараарай.

## Хэрэглэгчийн профайлын дэлгэрэнгүй мэдээллийг өргөтгөх

**Стори оноо: 3**

### Хэрэглэгчийн түүх:

Хэрэглэгчийн дэлгэрэнгүй мэдээллийг үзэх үед хэрэглэгчийн дэлгэрэнгүй хуудсанд дараах зүйлийг тусгасан байх ёстой:

- Хамгийн сүүлд байршуулсан зураг. Зургийн жижиг thumbnail зураг, байршуулсан огноо байх ёстой.
- Хамгийн олон сэтгэгдэлтэй зураг. Зургийн жижиг thumbnail зураг байх ёстой бөгөөд сэтгэгдлийг тоо байх ёстой.
- Эдгээр зургуудын аль нэг дээр дарснаар таны өмнө хэрэгжүүлсэн хэрэглэгчийн зургийн харагдац руу шилжих хэрэгтэй. Зураг нь бүх сэтгэгдэлтэй нь харагдах эсэхийг шалгаарай. Нарийн харагдах байдал нь та Төслийн 5 Нэмэлт Оноо хэсгийн фото stepper хэрэгжүүлсэн эсэхээс хамаарна.

### Зөвлөмж:

- Та өөрийн серверийн API дуудлагыг бичих хэрэгтэй болно. Мэдээллийн боловсруулалт нь урд талын хянагч руу орох ёсгүй.

## Сэтгэгдэлд @ меншион хийх

**Түүхийн оноо: 6**

### Хэрэглэгчийн түүх:

Хэрэглэгч зурагны сэтгэгдэлд хэрэглэгчдийг @меншион хийх, тухайн хэрэглэгчийг @меншион хийсэн бүх зургийг харах боломжтой байх ёстой

- Одоо байгаа хэрэглэгчийн @меншион хийхийг зөвшөөрөхийн тулд сэтгэгдэл нэмэх дэмжлэгийг сайжруулна уу. Тайлбар

бичих талбарт хэрэглэгчдийг сонгох ухаалаг арга байх ёстой (Зөвлөмжийг үзнэ үү).

- Хэрэглэгчид хүчингүй (байгаагүй) хэрэглэгчийг @меншион хийж чадахгүй байх ёстой бөгөөд хүчингүй хэрэглэгчийн хувьд @меншион хийх нь программыг гэмтээхгүй байх ёстой.
- Хэрэглэгчийг @меншион хийсэн сэтгэгдэл бүхий бүх зургийн жагсаалтыг харуулахын тулд хэрэглэгчийн дэлгэрэнгүй харагдацыг өргөтгөх, эсвэл @меншион байхгүй тохиолдолд боломжийн зүйлийг харуулах.
  - Зургийн жижиг thumbnail зураг. Зурган дээр дарснаар хэрэглэгчийн зургийн хуудсан дээрх зургийн байршилтай холбогдоно
  - Зураг эзэмшигчийн нэр. Эзэмшигчийн нэр дээр дарснаар эзэмшигчийн хэрэглэгчийн дэлгэрэнгүй хуудас руу холбогдох ёстой.

#### **Зөвлөмж:**

- React Mentions: <https://github.com/signavio/react-mentions> (React)

@меншион хийхийг сонгосон сэтгэгдэл болон сэтгэгдэл илгээх хооронд ялгаа байдгийг анхаарна уу. Сонголт дээр сэтгэгдэл бичих хэсэгт @меншион тэмдэглэх аргыг бодож үзээрэй.

Та тодорхой зурагтай холбоотой @меншион хадгалахын тулд мэдээллийн санг өөрчлөх шаардлагатай болно. Зурганд @меншион нэмж, @меншион хийсэн хэрэглэгчийн бүх зургийг буцаах API дуудлагатай байх нь ашигтай байж болох юм.

## **Зургийн харагдах байдлыг хянах**

**Стори оноо: 4**

**Хэрэглэгчийн түүх:**

Хэрэглэгч зураг байршуулахдаа аль хэрэглэгчид зураг болон түүний сэтгэгдлийг харахыг хязгаарлах боломжтой байх ёстой. Хэсэг нь дараах байдлаар ажиллах ёстой.:

- Хэрэглэгч зураг байршуулах үед тухайн зургийг харах боломжтой хэрэглэгчдийн жагсаалтыг (хуваалцах жагсаалт) зааж өгөх сонголтыг өгөх ёстой. Жагсаалт заагаагүй бол (жишээ нь, хэрэглэгч жагсаалт тодорхойлох сонголтоос татгалзвал) зураг нь хүн бүрт харагдах ёстой.
- Хэрэв зураг нь түүнтэй холбоотой хуваалцах жагсаалттай бол зөвхөн тухайн зургийн эзэмшигч болон жагсаалтад байгаа хүмүүс зургийг харах боломжтой. Хэрэв тухайн хэрэглэгч хуваалцах жагсаалтад ороогүй бол нэвтэрсэн хэрэглэгчдэд үзүүлэх зургийн тоонд үүнийг харуулах ёсгүй.
- Хуваалцах жагсаалт хоосон байвал зөвхөн тухайн зургийг эзэмшигч л харж болно. Энэ нь "жагсаалт заагдаагүй(no list is specified)" -ээс ялгаатай гэдгийг анхаарна уу.

#### **Зөвлөмж:**

- Зургийн холбогдох зөвшөөрлийг хадгалахын тулд та схемээ өөрчлөх шаардлагатай болно.
- Та зөвхөн хэрэглэгч үзэх зөвшөөрөлтэйг нь буцааж өгөхийн тулд зураг буцаах асуулгыг өөрчлөх шаардлагатай (жишээ нь, front end хянагч (controller) дахь зургийг шүүж болохгүй).

## **Үйл ажиллагааны урсгал**

### **Стори оноо: 4**

#### **Хэрэглэгчийн түүх:**

Хэрэглэгч зураг хуваалцах сайт дээр болсон үйл ажиллагааны жагсаалтыг харах боломжтой. Хэсэг нь дараах шаардлагыг хангасан байх ёстой.

- Toolbar дээрх "Үйл ажиллагаа" гэсэн товчлуур нь хэрэглэгчийг тухайн вэб сайт дээр болсон хамгийн сүүлийн 5 үйлдлийг он цагийн урвуу дарааллаар (хамгийн сүүлийнх нь дээд талд) харуулсан харагдуулна. Үзүүлсэн үйлдлүүд нь өвөрмөц байх албагүй гэдгийг анхаарна уу (жишээ нь, 5 зураг байршуулах үйл ажиллагаа байж болно).
- Үйл ажиллагаа бүр нь үйл ажиллагааны огноо, цаг, үйл ажиллагааг гүйцэтгэсэн хэрэглэгчийн нэр, үйл ажиллагааны төрөл (жишээ нь: Зураг байршуулах, шинэ сэтгэгдэл гэх мэт) болон үйл ажиллагааны тодорхой мэдээллийг дараах байдлаар харуулах ёстой.
  - Зураг байршуулах - зургийн жижиг thumbnail зургийг харуулах
  - Сэтгэгдэл нэмэх - сэтгэгдэл үлдээсэн зургийн жижиг thumbnail зургийг харуулах
  - Хэрэглэгч бүртгүүлэх - нэмэлт мэдээлэл байхгүй
  - Хэрэглэгч нэвтэрч байна - нэмэлт мэдээлэл байхгүй
  - Хэрэглэгч гарах - нэмэлт мэдээлэл байхгүй
- Үйл ажиллагааны дэлгэцийн харагдац нь шинэ үйл ажиллагаа гарах үед өөрийгөө шинэчлэх эсвэл шинэчлэх товчлуурыг өгөх ёстой бөгөөд ингэснээр хэрэглэгч шинэчлэлтийг идэвхжүүлнэ.

#### **Зөвлөмж:**

- Гүйцэтгэсэн үйл ажиллагааны бүртгэлийг хадгалахын тулд та мэдээллийн сангийн схемийг өргөтгөх шаардлагатай болно. Үүнийг олон янзаар хийж болох боловч MongoDB дээр бичлэгийн газрын (record maps) зураг бүрийг сайн хадгалах шинэ объект үүсгэх арга юм. Үйл ажиллагааны түүхийг тооцоолохдоо үйл ажиллагааны бичлэгийн объектууд дээрх асуулгын тусламжтайгаар хийж болно.
- Та өмнө нь байгаа сэтгэгдэл, зургуудын тэжээлд оруулгатай байх шаардлагагүй (in the seeded database)

## Сэтгэгдэл, зураг, хэрэглэгчдийг устгах

**Стори оноо: 4 оноо**

### **Хэрэглэгчийн түүх:**

Хэрэглэгч сайт дээрх өөрийн эзэмшдэг аливаа зүйлийг устгах боломжтой байх ёстой:

- Түүний аль ч зураг
- Түүний хийсэн ямар нэгэн сэтгэгдэл (үүнд бусад хэрэглэгчдийн зураг дээр бичсэн сэтгэгдэл багтана)
- Бүх хэрэглэгчийн бүртгэл. Хэрэв хэрэглэгч өөрийн бүртгэлээ устгахаар шийдсэн бол эцсийн анхааруулга гарч, дараа нь хэрэглэгч нэвтэрч, мэдээллийн сангаас түүний бүх мэдээллийг устгана. Энэ нь та хэрэглэгчийн объектыг устгахаас гадна энэ хэрэглэгчийн үүсгэсэн бүх зураг, сэтгэгдлийг устгах хэрэгтэй гэсэн үг юм. Үндсэндээ бид хэрэглэгчийг анхнаасаа хэзээ ч байгаагүй гэж үзэх болно. Хэрэгжүүлсэн бусад түүхүүдэд нэмсэн бусад зүйлсийг (жишээлбэл, үйл ажиллагаа, харагдах байдал, меншион гэх мэт) мөн тусгах ёстой.

### **Зөвлөмж:**

- Хэрэглэгч янз бүрийн объектыг устгах үед та мөн устгах шаардлагатай байж болох холбоотой мэдээллийг анхаарч үзэх хэрэгтэй. Энэ нь сервер дээр хийгдэх бөгөөд та холбоо бүхий объектуудыг хэрхэн зөв устгах талаар судалгаа хийх шаардлагатай болж магадгүй юм.
- Хэрэглэгч өөрийн эзэмшдэггүй зүйлийг устгах боломжгүй байх ёстой тул үүнийг бас нягтлаарай.

## Зурагт “like” дарах

**Стори оноо: 4 оноо**

### **Хэрэглэгчийн түүх:**

- Facebook-ийн дуртай товчлууртай адилаар хэрэглэгчдэд зураг таалагдах боломжийг олгохын тулд зураг бүрт “like” товчийг хэрэгжүүлээрэй(товчлуур, [icon](#), гэх мэт.).
- “like” товчлуур дээр дарснаар хэрэглэгч зураг дээр like дарсан байх болно (мөн like дарсан товчлуурыг буцаан like-ийг буцаан авахаар дүрслэн харуулна). Нэг хэрэглэгч аль хэдийн таалагдсан зурган дээрх "Like" товчийг дарснаар тухайн зураг таалагдахгүй буюу like-ийг буцааж байх ёстой.
- Зурганд таалагдсан хүний тоо нийт хэрэглэгчдийн тооноос хэзээ ч их байж болохгүй.
- Хэрэв хэрэглэгч аль хэдийн зураг таалагдсан бол тухайн хуудсан дээр тухайн зураг хэрэглэгчдэд аль хэдийн таалагдсан гэсэн харагдац бүхий тэмдэг байх ёстой. Хэрэв хэрэглэгч зураг таалагдаагүй бол энэ дүрсийг арилгах хэрэгтэй.
- Таалагдсан товчлуурын хажууд тухайн зурагт таалагдсан тоог харуулах ёстой. Зурган дээр таалагдсан эсвэл таалагдаагүй бол энэ тоог нэн даруй (харагдахуйцаар) шинэчлэх шаардлагатай.
- Хэрэглэгчийн зургийн хуудсыг таалагдсан тоогоор нь буурах дарааллаар эрэмбэлсэн байх ёстой (хамгийн их таалагдсан зургууд дээд талд байна). Хэрэв зургууд ижил тооны лайктай бол цаг хугацааны тэмдгээр урвуу дарааллаар эрэмбэлнэ үү (хамгийн сүүлд эхнийх). Хэрэглэгч зураг таалагдсан эсвэл таалагдаагүй тохиолдолд үүнийг нэн даруй өөрчлөгддөг байх шаардлагагүй бөгөөд- хуудсыг дахин шинэчлэх хүртэл хүлээгддэг байна.

#### **Зөвлөмж:**

- Та зурагтай холбоотой лайк болон лайк үүсгэсэн хэрэглэгчдийг оруулахын тулд схемийг нь шинэчлэх шаардлагатай болно.
- Та лайк бүр хүчинтэй эсэхийг баталгаажуулахын тулд сервер дээр баталгаажуулалт хийх шаардлагатай болно. Мөн уралдааны нөхцөл байдлыг ( [race conditions](#) ) судалж, энэ нь

том хэмжээний системд ямар нэгэн асуудал үүсгэж болзошгүй эсэхийг тодорхойлно уу.

## Хэрэглэгчид таалагдсан зургийн жагсаалт

### Стори оноо: 6

#### Хэрэглэгчийн түүх:

Нэвтэрсэн хэрэглэгч дуртай зургуудаа үзэх боломжтой байх ёстой бөгөөд шинэ хуудсан дээр хэрэглэгчийн тусгайлсан дуртай жагсаалт гарч ирэх ёстой (зам: /favorites). Энэ хуудас нь тухайн хэрэглэгчийн дуртай зургуудын жагсаалтыг харуулахад зориулагдсан байх ёстой. Хэрэглэгчид энэ жагсаалтаас дуртай зургуудаа устгах боломжтой байх ёстой.

Хэрэглэгчид дурын зургийн жагсаалтад харагдах дурын зургийн хажууд гарч ирэх товчлуур дээр дарснаар дурын зургийг дуртай гэж тэмдэглэх боломжтой байх ёстой. Зургийг дуртай зургаар сонгохын тулд дараах шаардлагыг хангах ёстой:

- Хэрэглэгчийн зургийн жагсаалтад тухайн зургийг дуртай болгох товчлуурыг идэвхгүй болгох хэрэгтэй.
- Энэ зургийг одоогоор хэрэглэгчид таалагдсан буюу дуртай гэж үзсэн тодорхой үзүүлэлт байх ёстой.

Шинэ харагдацын компонент хэсэг дээр хэрэглэгч дараах үзүүлэлтүүдийг хангасан таалагдсан зургийн жагсаалтыг харах боломжтой байх ёстой:

- Зураг бүрийг жижиг thumbnail зургаар дүрсэлсэн байх ёстой.
- Thumbnail зураг дээр дарснаар зургийн илүү том хувилбарыг харуулах модаль цонх [modal window](#) нээгдэнэ (доорх зөвлөмжийг үзнэ үү). Зөвхөн товшилтоор хэрэглэгчийг зөвхөн тухайн зургийг агуулсан шинэ хуудас руу чиглүүлэх нь хангалтгүй юм.

- Модал нь зурагтай холбоотой огнооноос бүрдсэн тайлбарыг агуулсан байх ёстой.
- Жагсаалтаас мэдрэмжээрээ зургийг устгах (жишээ нь, thumbnail зургийн баруун дээд "x" дээр даргах) хялбар арга байх ёстой бөгөөд ингэснээр хэрэглэгч хуудсыг дахин сэргээсэн болон хаягаас гарсан ч гэсэн энэ нь хэрэглэгчийн таалагдсан жагсаалтад цаашид орохгүйгээр байлгах болно.

Өөр дурдах зүйлс:

- Дуртай зургуудын жагсаалтад url зам /favorites (#!/favorites эсвэл #/favorites) хаягаар хандах боломжтой байх ёстой.
- Хэрэглэгч нэвтэрсэн эсэхээс шалтгаалж, дэлгэцийг хажуу талд (sidenav) эсвэл дээд хэсгийн самбараас (toolbar) таалагдсан хуудсанд нөхцөлтэйгээр харуулна.
- Дуртай зургуудын энэ жагсаалт нь олон сессийн турш хадгалагдах ёстой: өөрөөр хэлбэл, та энэ жагсаалтыг хянахын тулд сервер талын мэдээллийн санг ашиглах ёстой бөгөөд хэрэглэгчид өөрсдийн дуртай хуудсанд зочлох үед жагсаалт үүсгэхийн тулд энэ мэдээлэлд хандах ёстой.

**Зөвлөмж:**

React Hints

- Та <https://github.com/reactjs/react-modal>, <https://mui.com/material-ui/react-modal/>, эсвэл <https://github.com/react-bootstrap/react-overlays> хэрэгцээтэй гэж бодох байх. Бүрэлдэхүүн хэсгүүдтэй номын санг (component with library) ашиглахдаа баримт бичгийг анхааралтай уншина уу.
- Энэ түүхийн томоохон сорилт бол таны зураг хуваалцах аппликейшнд хамаарлыг (dependencies) суулгаж, зөв тохируулах явдал юм.
- `npm install`-ын үед ямар ч **UNMET PEER DEPENDENCY** `<package-name>` нь одоогийн багцаа суулгахаасаа өмнө `<package-name>`-г суулгах шаардлагатай гэсэн үг юм.

# Хэрэглэгчдийн шинэ үйл ажиллагааг тэмдэглэдэг хажуугийн цэс

Стори оноо: 3

## Хэрэглэгчийн түүх:

Хажуу талын навигацийн цэсэн дэх хэрэглэгчдийн жагсаалтыг өргөтгөн, зөвхөн нэвтэрсэн хэрэглэгчийн биш, бүх хэрэглэгчийн хамгийн сүүлд хийсэн үйлдлийг харуулдаг болгоно уу. Үйл ажиллагаа гэдэгт дараах зүйлсийг хамааруулна:

**Хэрэглэгч** (зөвхөн нэвтэрсэн хэрэглэгч биш) бүрийн хийсэн хамгийн сүүлийн үйлдлийг харуулахын тулд хөтөчийн хажуугийн цэс дээрх хэрэглэгчийн жагсаалтыг сунгах ба үйл ажиллагаанд дараах хэсгүүд орно:

- зураг нийтэлсэн **\*\*зургийн дэлгэцний зураг (thumbnail) шаардлагатай\*\***
- сэтгэгдэл нэмсэн
- хэрэглэгчээр бүртгүүлсэн
- хэрэглэгч нэвтэрсэн
- хэрэглэгч гарсан

Дэлгэц автоматаар шинэчлэгдэх ёстой.

## Зөвлөмж:

- Одоогийн хэрэглэгч хэн болох, түүний сүүлийн үеийн үйл ажиллагаа, найз нөхдийн жагсаалтыг сүүлийн үеийн бүх үйл ажиллагаатай нь тодорхой ялгахын тулд хажуугийн самбар дээрх хэсгийг дахин шинэчлэн засварлахыг зөвлөж байна.

# Зураг тааглах

Стори оноо: 10 оноо

## Хэрэглэгчийн түүх:

Хэрэглэгч зураг дээр хүмүүсийг таглах боломжтой. Энэ шинэ функц хэрхэн ажиллах ёстойг харуулсан бүдүүвч зураг эдгээрээс тогтоно:

- Хэрэглэгчид тухайн зургийн тэгш өнцөгт хэлбэртэй хэсгийг сонгож, тухайн хэсэгт байгаа хүнийг таних замаар зургийг таглаж таних болно.
- Зурган дээр хулганыг чирж тэгш өнцөгтийг сонгоно. Хэрэглэгч тухайн хэсэг дээр дарах үед тэгш өнцөгт нь сунаж, багасах ёстой. Хэрэглэгч ямар ч чиглэлд чирэх боломжтой байх ёстой бөгөөд чирж дууссаны дараа тэгш өнцөгт харагдах харагдац хэвээр байх ёстой.
- Хэрэв хэрэглэгч тэгш өнцөгтийг зургийн хэмжээнээс гадагш сунгахыг оролдвол та тэгш өнцөгтийг зургийн хэмжээ хүртэл л сунгах боломжтой.
- Сонгосон тэгш өнцөгт нь хэрэглэгчдэд таалагдахгүй бол түүнийг солихын тулд өөр тэгш өнцөгтийг гарган сонгох боломжтой.
- Хэрэглэгч зургийн хэсгийг сонгосны дараа системд бүртгүүлсэн хэрэглэгчдээс нэр сонгож, хэрэглэгчийг таглах боломжтой байх ёстой. Таг нь илгээгдэж, мэдээллийн санд хадгалагдана.
- Нэг зурагт олон таг байж болох бөгөөд тус бүр өөрийн гэсэн тэгш өнцөгт болон хэрэглэгчийн нэртэй байдаг.
- Аппликешн нь зургийг бүх тагыг нь харуулах механизмыг хангах ёстой: таг бүрийн тэгш өнцөгтийг зураг дээр харуулах бөгөөд хулганыг тэгш өнцөгт дээр аваачихад таг нь хэрэглэгчийн овог, овог нэрийг харуулах ёстой.
- Хэрэглэгчийн нэр дээр дарснаар таглагдсан хэрэглэгчийн хэрэглэгчийн дэлгэрэнгүй харагдац руу шилжих хэрэгтэй.

### **Зөвлөмж:**

- **Гүйцэтгэхэд хэцүү тул энэхүү даалгаврын оноо нь 10 болно!**  
Front end нь аль нэг талбар сонгох модулийг хайж олох, түүнийг ажиллуулах эсвэл өөрийн талбай сонгох механизмыг бий болгох шаардлагатай..
- Зураг бүр нь зурагтайгаа холбоотой байрлал дээр тагтай байх ёстой. Энэ харьцангуй (relative) байрлалыг ашиглан тагуудыг зөв харуулах боломжийг танд олгоно.
- Байршлын тооцоолол нь цонхны гүйлгэх байрлалд мэдрэмтгий биш гэдгийг баталгаажуулахын тулд хуудсыг өөр газар гүйлгэж байх үед зургуудыг таглахыг туршиж үзээрэй.
- Зарим гуравдагч талын номын сангууд (library) танаас зургийн өргөн ба/эсвэл өндрийг өгөхийг шаардаж магадгүй.