School of Information
Technology and Electronics



ICSI301 WEB APPLICATIONS

PhD. AMARSANAA Ganbold

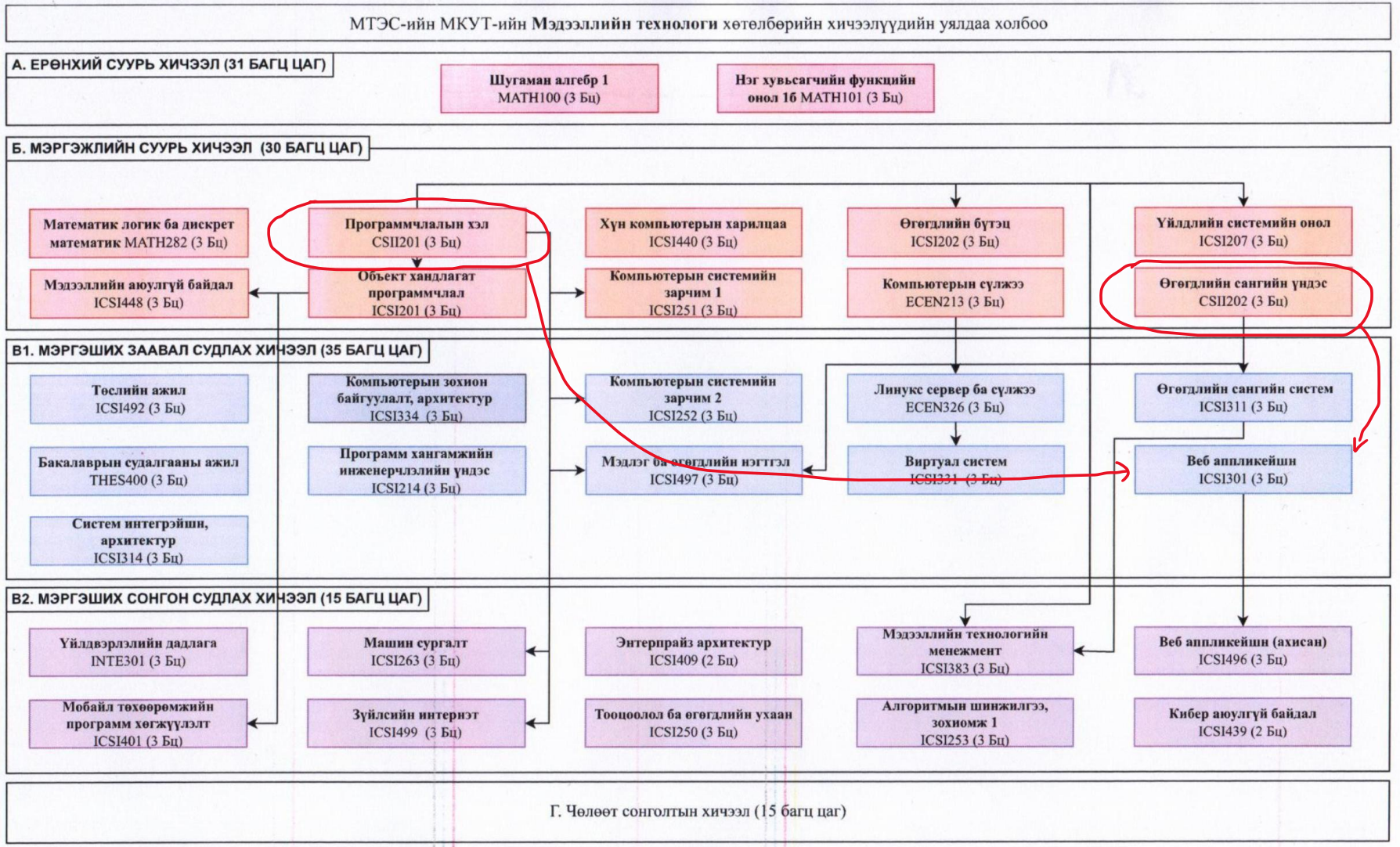
DEPARTMENT OF INFORMATION AND COMPUTER SCIENCE
School of Information Technology and Electronics, NUM
amarsanaag@num.edu.mn

2026 Fall

General information

- Major compulsory course for IT, SW & IS programs
- **Credit hours**: 3
  - Weekly
    - 2 hours of lecture
    - 3 hours of lab session
- **Course schedule**: 16 weeks
- **Instructors**:
  - Prof. Amarsanaa Ganbold (lecture, labs)
  - Sarnai Gantulga (labs)

PREREQUISITES



ICSI301 is a follow up course of

CS142 - Web Applications

http://cs142.stanford.edu

Mendel Rosenblum
mendel@cs.stanford.edu

Gmail

Amarsanaa Ganbold <amarsanaag@gmail.com>

NEED APPROVAL: use of CS142 lecture notes

Mendel Rosenblum <mendel@cs.stanford.edu> Wed, Sep 4, 2019 at 2:31 AM
To: Amarsanaa Ganbold <amarsanaag@seas.num.edu.mn>

I'm happy to have you use my lecture notes.

Mendel

On Tue, Aug 27, 2019 at 10:15 PM Amarsanaa Ganbold <amarsanaag@seas.num.edu.mn> wrote:

> Dear prof. Mendel Rosenblum,
>
> I am a faculty in the CS department of the National University of Mongolia and I have been following your wonderful CS142 course content for the past three years. Can I use lecture notes of the course in my web programming course? It would be great if you approve.
>
> Sincerely,
> --
> Amarsanaa Ganbold,
>
> Department of Information and Computer Science,
> School of Engineering and Applied Sciences,
> National University of Mongolia

Approved by prof. Mendel Rosenblum

Today: FAQ

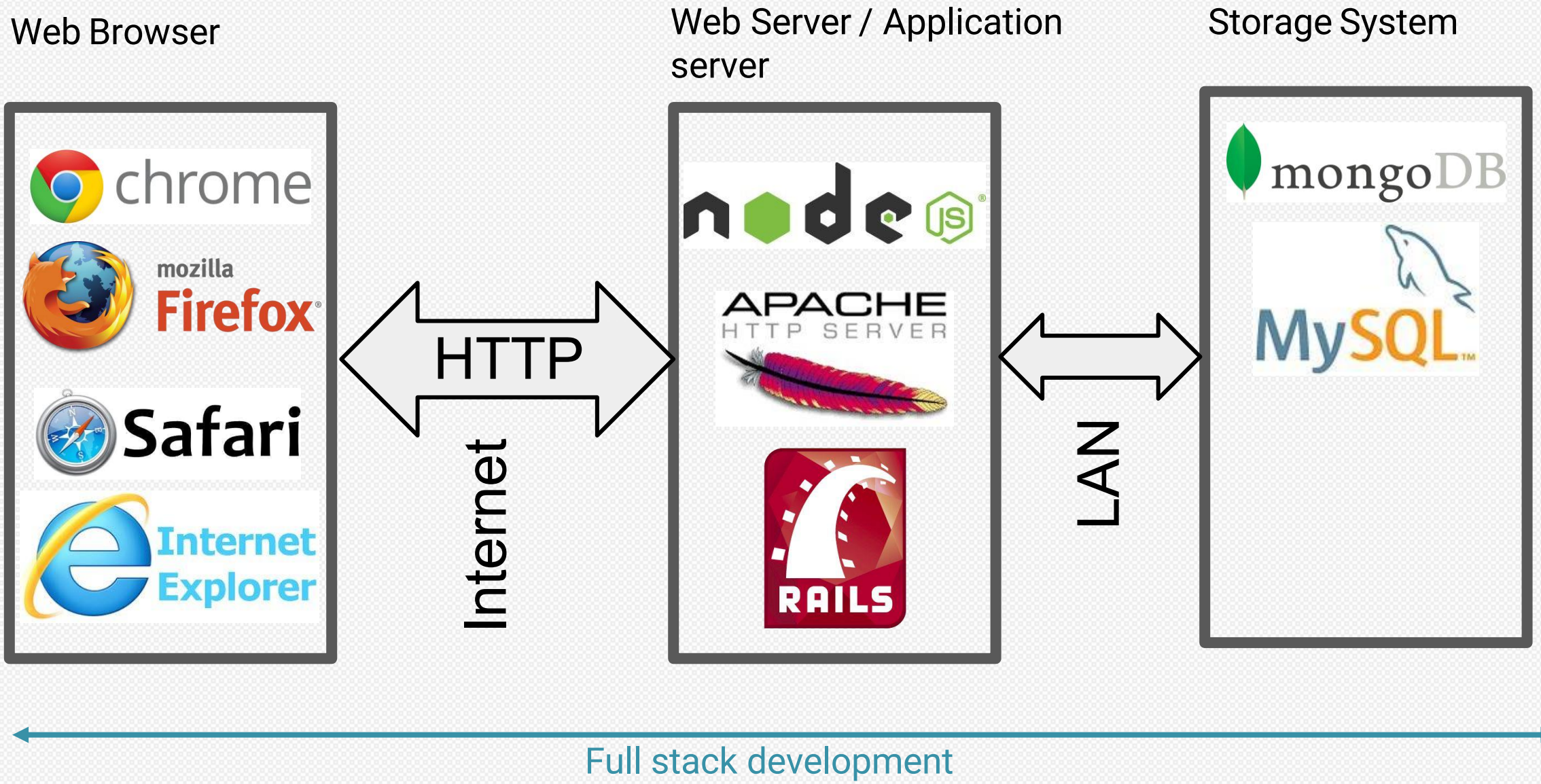
- What is this course about?
- How is my course grade determined?
- Who is teaching the course?
- How do I communicate with the course staff?
- What kind of programming projects will I have do?
- What kind of computing environment do I need?
- Do I need to buy a textbook?
- Are the course lectures record on video?

Course is about Web Applications

- Technologies used to **build** modern web applications
  - Note: CS14x (computer systems course in Computer Science department)
  - Full stack: Browser ⇔ Web server ⇔ Storage system
  - Goal: Learn how a web application is built and run

- How to build a web application - learn by doing:
  - Use MERN stack (React.js, Node.js, Express.js, MongoDB)
  - Goal: Build a photo sharing web app and understand how it works!

Full Stack Web Application Architecture



CS142 Technologies and Concepts

- Browser environment:
  - HTML/CSS/JavaScript - Markup, separation of content & style, reuse
  - Document object Model (DOM) - Document structure
- Browser software:
  - Model View Controller, Single page applications, Responsive design - React.js
- Backend communication:
  - API design - HTTP/AJAX/REST/GraphQL
  - Cookies/Sessions - Storage/Trust
- Backend implementation:
  - Web Server - HTTP request processing - Node.js
  - DBMS - Schema, Objects, CRUD, indexes, transactions - MongoDB
  - End-to-End - Scale and Security

Grading

40% Projects - 8 projects (usually bi-weekly, due date lab session)

- Projects 1-4: Learn technologies in front-end: HTML/CSS/React.js
- Projects 5-8: Building a Photo Sharing App using React.js/Node.js/MongoDB

20% Mid-term Exam

- Test

30% Final Exam

- Closed book, with limited note pages
- Note: End-Semester examination period

10% Attitude

- Active presence & participation in class (e.g., asking question, respect others)
  - No presence: -1 point
- No violations Code of Honor (not giving improper assistance)
- Posting a discussion or answering them

Course Material and Grading

- CS142 is different from introductory programming class
- Lectures cover many more **concepts** than are addressed in the programming projects
  - Lecture focused on concepts, not directly helping with project coding
- Exams focused on **concepts** presented in class but not used in projects
  - Possible to do well on all the projects and not get an A in the class
  - Need understanding beyond "magic incantations"

Books & Materials

- JavaScript: The Good Parts by Douglas Crockford (O'Reilly Media).
  - As the book's title suggests, JavaScript is not a simple language and it is easy to hurt yourself with some of its features. This book recommends a somewhat more safe subset of the language to use.
- JavaScript Patterns by Stoyan Stefanov (O'Reilly Media).
  - JavaScript programmers have discovered ways around issues in the language by doing things in certain ways (i.e. patterns). These patterns are commonly used in JavaScript programs but some of them use strange features of the language so it is not clear what is going on if you don't recognize them as a common pattern.
- JavaScript: The Definitive Guide, 6th Edition by David Flanagan (O'Reilly Media).
  - As the title suggests this is a comprehensive description of JavaScript.
- Eloquent JavaScript: A Modern Introduction to Programming, 2nd Edition by Marijn Haverbeke.
  - A free e-book.
- ECMAScript® 2016 Language Specification
  - The JavaScript standard document.
- MDN (Mozilla Developer Network)
  - a documentation repository and learning resource for web developers by developers

Communication

- **CS142 course website - http://cs142.stanford.edu**
  - https://kde-milab.github.io/cs142-website-mn/projects.html
- **Microsoft Teams group**
  - everyone can post a question and reply (don't be shy, everyone is learning)
  - Online lectures will be recorded temporarily as Teams provided
  - online communication protocol:
    - Lectures will be asynchronous
    - Raise a hand in Teams meeting
    - Ask a common question via a post in a channel, so let others know
    - Specific questions, chat or ask (via online call) during lab session
- **In-person meetings (office hours) – NUM Building #8, No. 200B**
  - Could be replaced by online meeting (amarsanaag@num.edu.mn or Teams chat)
  - By appointment only

Project details

1. HTML & CSS
2. JavaScript
3. Browser Document Object Model (DOM)
4. Learn React.js - Single page application
5. **Photo Sharing App**
6. **Backend server - Node.js and MongoDB**
7. **Sessions state and validation**
8. **Photo App Scrumboard**

Deadline: Monday, 1:00 PM

Class software requirements

- A modern web browser
  - Chrome is strongly suggested
- Node.js
  - Installs fairly easily on modern OS environment (Linux, MacOS, Windows)
  - npm (in Node.js install) is used for fetching assignments and dependencies
- MongoDB
  - Easy to install (for a DBMS) on modern OS environments

- Code Editors
  - Visual Studio Code
  - Atom
  - Sublime Text

Student Dilemma of 2026

- AI is advancing extremely rapidly and is the focus of so much industry and research
- Employers demand AI expertise, but you can't responsibly create production-ready code with AI alone. You still have to master fundamental concepts and problem-solving skills
- Improperly using AI in your fundamental courses will **obliterate** your ability to learn, so where does AI fit into your education?

Source: CMU 15-113 Effective Coding with AI

Student Dilemma of 2026 (2)

- Plus, the job market sure feels like an absolute wasteland…
- To stand out, you need:
  - Fundamental knowledge and well-honed problem-solving skills
  - Experience with incorporating AI into your workflow efficiently and responsibly
  - Interesting projects and experiences

Source: CMU 15-113 Effective Coding with AI

Stanford Honor Code

- We want you to do the projects individually, for future team member

- "use, without giving reasonable and appropriate credit to or acknowledging the author or source, of another person's original work, whether such work is made up of code, formulas, ideas, language, research, strategies, writing or other form.", *Stanford University, CS and the Honor Code*
  - **Rule 1: You must not look at solutions or program code that are not your own.**
  - **Rule 2: You must not share your solution code with other students.**
  - **Rule 3: You must indicate on your submission any assistance you received.**

- Avoiding plagiarism
  - You must write your own code
  - Do not share code or write code collaboratively
  - It's fine to discuss overall strategy, share tips about Web technologies (useful CSS styles, library methods, etc.)

Project details

1. HTML & CSS
2. JavaScript
3. Browser Document Object Model (DOM)
4. Learn React.js - Single page application
5. **Photo Sharing App**
6. **Backend server - Node.js and MongoDB**
7. **Sessions state and validation**
8. **Photo App Scrumboard (only the project that vibe coding allowed)**
   - Submit the prompts and dialogues from your sessions
     - At the top right corner of the Chat-GPT interface, there is a “share” option that allows you to create a link to your dialogue
   - Submit a 200-word reflection on your experience (written without AI help)

Inspired by the **CSM.100 policy**

Any questions?