Welcome to 18652 - Foundations of Software Engineering (FSE)

18-652 is a central course in the MS-SE program. It is a prerequisite for most courses in the 18-65X series. Below is the information to prepare you for the course and facilitate ramp-up.

Before you Attend the First Class

- ❏ Learn JavaScript and Node.js.
- ❏ Complete the Pre-Registration Programming Assignment and video.
- ❏ Learn Git and obtain a Github account.

Textbooks and Readings

There are no mandatory textbooks. Necessary readings will be provided throughout the course.

Programming Skills

We will assume that you have significant experience with a modern programming language. You will need to know how to build a simple web application end-to-end and be familiar with programming in **JavaScript** and **Node.js** to complete the team project component. *To prepare you for the course, you will build a simple web application from scratch, using a specific development stack* ***before you take the course*****. *Refer to the Pre-Registration Programming Assignment below.***

Git & Github

You will use Git & GitHub for version control and issue tracking. *Obtain a GitHub account, complete a Git tutorial, review the Git reference, and experiment with Git.*
Git reference: http://gitref.org/
Here is a free tutorial on Git: https://www.codeschool.com/courses/try-git
If you need more or something different, Codeschool also has a free course on Git: https://www.codeschool.com/courses/git-real
A compact, no-frills starter guide to git is here: http://rogerdudler.github.io/git-guide/

18652 Pre-Registration Programming Assignment

The purpose of the assignment is to provide you with a context to demonstrate proficiency in a modern programming language, and to make sure that you are ready to tackle the course project. While programming is not the main focus of the course, a lot of programming is involved. You will use specific technologies and require proficiency in them. ***This assignment will be graded. It is due the first week of classes.***

Instructions

- Build a **web application** that implements the requirements described below.
- You will use the following development stack to complete this assignment:
    - Client side: the standard "web stack": **HTML, CSS, JavaScript**
    - Server side: **Node.js** with **express.js** web development framework
    - Database: any database, SQL or No-SQL, that runs on Linux.
    - You may additionally use other tools, libraries, node modules, frameworks, middleware, or plug-ins.
    - The use of **express.js** is mandatory. You may **not** substitute it with another web development framework.
- **Push your code to a GitHub repository, become comfortable using GitHub and git.**
- Create a **short demo video** (max 5 mins) for your application. Describe your application's **purpose, design**, **code structure**, choice of **technologies**, and demo your **application**.
- The user interface must be **web-based**: users will use the application through a **browser**.

Checklist

Here is a checklist for your Pre-Registration Programming Assignment:

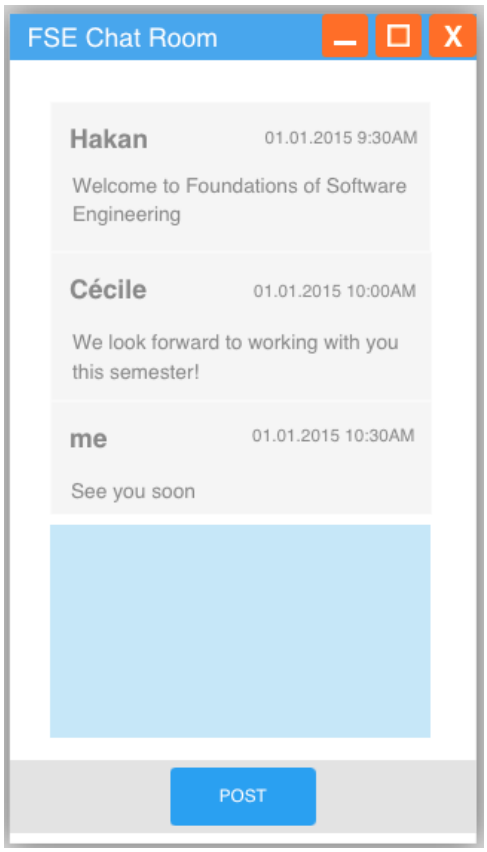
- ❏ Application has required look-and-feel.
- ❏ Application uses real-time dynamic updates to display new chat messages.
- ❏ Application satisfies its other functional requirements.
- ❏ Application uses a database and persists the data.
- ❏ A 5-min video is created and uploaded to youtube.
- ❏ Application design is described in video.
- ❏ Code structure is reviewed in video.
- ❏ Technologies used together with their purposes are explained in video.
- ❏ Application functionality is demonstrated in video.
- ❏ Your code is pushed to github.

You have completed the assignment if you can tick off all items on this list.

Demonstration and Code Review

We may ask you to demonstrate your running application and discuss your design and code at the beginning of the semester. We may review your code on github.

Application Requirements



Implement a simple FSE Chat Room application. The system should allow a user to:

- Enter the chat room with his/her name
- See other users' chat messages
- Post a chat message
- Leave the chat room

The application should satisfy the rules:

- When a user posts a chat message, the text is displayed together with the user's name and current timestamp.
- When there is a new post, the screens of all of the users in the chat room are instantly and **dynamically updated** (the updates are real time on all clients).
- All the chat messages should be stored on the server in a database and re-loaded when a user exits and re-enters the chat room.

A mockup of the chat room's main screen is shown on the left. Your application should have a similar look and feel.

Additional Resources

In addition to **express.js** (lightweight web application framework), you may find the following middleware and other JS libraries useful:

- jQuery (JavaScript library for HTML/json manipulation and traversal)
- Jade or EJS (Node.js template engine)
- underscore (JavaScript library for convenient functional programming constructs)
- socket.io (JavaScript-based middleware for real-time web applications)
- Bootstrap (client-side framework for generating responsive user interfaces)

Other alternatives are available to, but these are good standard choices.

Starter guides to JavaScript is here:

- http://www.w3schools.com/js/default.asp (with a cloud IDE to instantly try out examples)

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

Codeschool has decent online courses on JavaScript, Node.js, and HTML5/CSS. You may want to get a one-month subscription to access everything, and then move onto free online sources.

- https://www.codeschool.com/paths/javascript
- https://www.codeschool.com/paths/javascript#node-js
- https://www.codeschool.com/paths/html-css#css3

For express.js, here are some resources. Start with the guide. Then watch the screencasts. The third site gives a brief overview, and links to expressjs.com for details.

- http://expressjs.com/
- http://expressjs.com/2x/screencasts.html
- http://webapplog.com/express-js-fundamentals/

There are plentiful online resources and tutorial that will help you ramp up with Node.js and JavaScript. Most tutorials illustrate these technologies by showing how to develop a simple, real-time chat application.

Using Object Orientation (OO) with JavaScript (JS)

This will be really important during the course to give structure to your JS code. **In particular, your server-side code written in Node.js must be well structured**. A caveat of JS is that it allows many kinds of unsafe, unstructured coding practices that lead to “spaghetti code.” JS is a multi-paradigm language that supports OO, albeit with some effort. Keep your JS code structured by using objects properly and adhering to OO principles. Use the latest version of JS and Nodel.js, which support OO better with ES6 syntax. Again, this is especially important for your server-side code. Here are a few resources that may help you:

- http://www.w3schools.com/js/default.asp -- check the section on JS Objects
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript

Good luck!