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.

Textbooks

We'll use two textbooks:

- Henrik Kniberg. Lean from the Trenches: Managing Large-Scale Projects with Kanban
  Please read the entire book. It's an easy read.
- Bernd Bruegge, Allen H. Dutoit. Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition)
  Read Chapter 1, Chapter 2, and as much as possible from Part II.

Programming Skills

We will assume that you have significant experience with at least one object-oriented programming language. You will need to know how to build a simple web application end to end and be familiar with programming in **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 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

Linux

When deployed, your team project system will run on Linux. The web is full of Linux resources.
Start with one of these if you have never worked with Linux or Unix:

- http://ryanstutorials.net/linuxtutorial/
- http://www.ee.surrey.ac.uk/Teaching/Unix/

BeagleBone Black

Beaglebone Black is a neat system-on-a-chip board.
We will give each project team one to be used in the course project.
Find more about BeagleBone here if you're curious: http://beagleboard.org/.

Enjoy, and see you soon!

============ 18652 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.

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” triple: **HTML, CSS, JavaScript**
    - Server side: **Node.js** with **express.js** web development framework
    - Database: **SQLite**
  - You may additionally use other tools, libraries, modules, frameworks, middleware, or plug-ins.
- 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 design decisions, choice of technologies, and demo your application.
- The user interface must be **web-based**: users will use the application through a **browser**.

Demonstration

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

This assignment will be graded. It is due at the end of the first week of the course.

Application Requirements

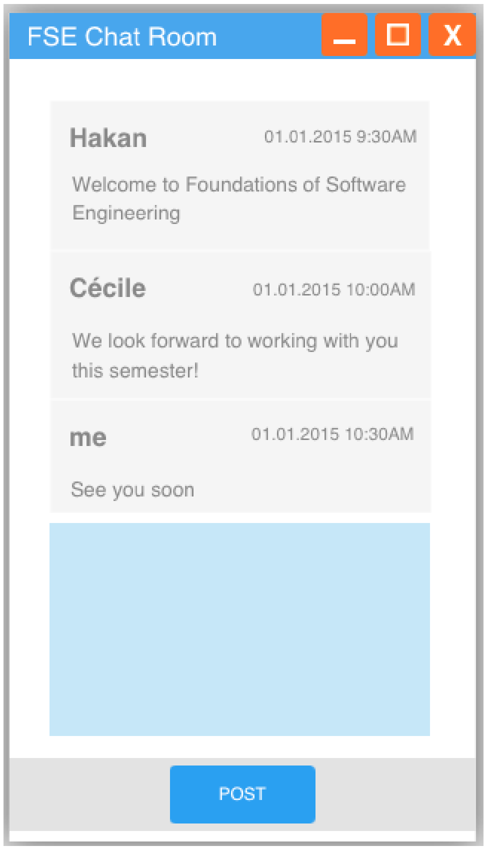
Implement a simple FSE Chat Room application. The system should allow a user to:

- 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 chat room is dynamically updated on all the screens of the users in the chat room (the updates are real time on all client browsers).
- All the chat messages should be stored on the server in a database and loaded when a user exits and re-enters the chat room.

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



Hakan 01.01.2015 9:30AM

Welcome to Foundations of Software Engineering

Cécile 01.01.2015 10:00AM

We look forward to working with you this semester!

me 01.01.2015 10:30AM

See you soon

Resources

In addition to express.js (light-weight web application framework), you may find the following middleware and other libraries useful:

- jQuery (JavaScript library for HTML manipulation and traversal)
- jade (Node.js template engine)
- socket.io (JavaScript-based middleware for real-time web applications)
- Bootstrap (client-side framework for generating responsive user interfaces)

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.