CMSC 100 2nd Sem 2018 Project Specs

Pick your own groupmates, maximum of 4 per group.

Objectives

For the project, students will create a backend API using NodeJS, and use ReactJS to create a frontend application that interacts with the backend server. They are required to use the following technologies:

1. NodeJS
2. ReactJS
3. MongoDB

Additional Guidelines:

- Be Creative
- You may use CSS libraries (Bootstrap, MDL Lite, etc.) I recommend react-bootstrap or react-semantic-ui because these can easily be integrated into react.
- Students may include any NPM package for their project EXCEPT:
    1. CMS packages

Grading Scheme

80% Functionality
20% User Interface

Presentation

Sometime before end of classes (May 18) (TBA)

Moodle

Create a moodle-like app that lets users sign up as a student or teacher. Students and teachers interact with app in different ways.

Functionality:

1. Account
    a. Sign up as either teacher or student
    b. Log in
    c. Non-logged in users can only view homepage/login page
2. User Type Specific
    a. Teacher
        i. Create/Edit/Delete Class
        ii. Create/Edit/Delete Post in Class
        iii. Delete Comment/Post made by Student in Class
        iv. Add/Remove Student to/from Class
        v. View List of Students in Class
        vi. Edit Class Settings
            1. Students can Post and Comment
            2. Students can only Comment
            3. Students cannot Post or Comment
    b. Student
        i. Join/Leave class
        ii. View List of classmates in Class
        iii. Create/Edit/Delete Post (if allowed by class settings)
        iv. Create/Edit/Delete Comment in Post (1 level deep only)
3. Inbox (All user types)
    a. Send private message to user (any user)
    b. View/Delete private messages in user inbox
    c. Mark private message as ‘read’ when viewed
    d. Reply to user from inbox
    e. Search for message by title, read/unread status
4. Search
    a. Search Class by Title
    b. Search Post within Class

Data

1. User
    a. Email
    b. Name
    c. Password
    d. UserType
2. Class
    a. Title
    b. Section
    c. Posts
    d. Students
3. Post
    a. Author
    b. Content
    c. Timestamp
    d. Comments

4. Comment
    a. Author
    b. Content
    c. Timestamp
    d. Like count
5. Message
    a. Sender
    b. Recipient
    c. Content
    d. Timestamp

Social Media

Create a social media app that lets users sign up, create a profile, post, comment, and add other users as friends.

Functionality

1. Account
    a. Create Account

b. View/Edit own profile (contains: Friends List, Name, About, Birthday)
c. Non-logged in users can only view homepage/login page

2. Post/Comment
   a. Create/Edit/Delete Post on own Wall
   b. Create/Edit/Delete Post on Friend’s Wall
   c. Create/Edit/Delete Comment on own Post (1-level deep only)
   d. Create/Edit/Delete Comment on Friend’s Post (1-level deep only)
   e. Like/Unlink Post or Comment (only one Like per User, per Post/Comment)
3. Notification
   a. Receive Notifications about:
      i. New Friend Request
      ii. New Post on Page (from friend)
      iii. New comment on Post
      iv. Likes in Posts/Comments
   b. Direct user to relevant page when clicking notification
   c. Mark notifications Read when clicked/viewed
4. Friends
   a. Add/Remove Friend (from profile page, or search result)
   b. Approve/Reject Friend Request
   c. View Friend’s Posts, Profile Page
5. Search
   a. Search for User by name
   b. View User’s (not a Friend) Profile, but not Posts
6. Settings
   a. Unsearchable (profile does not appear in search, but can be visited in direct URL link)
   b. Not accepting friend requests

Data

1. User
   a. Email*
   b. Name*
   c. Password*
   d. About
   e. Birthday

2. Post
   a. Author
   b. Content
   c. Timestamp
   d. Comments

    e. Like count

3. Comment
    a. Author
    b. Content
    c. Timestamp
    d. Like count