1 Summary

You and your group are required to build a functional web application that makes use of the technologies and techniques we worked on in the first half of the class. As I consider this project to make up the entire second half of the course, it is worth 50% of your course grade. This will be distributed across multiple checkpoints to ensure you are making progress and provide feedback on how you are doing. There will be three separate group project assignments due between now and Thanksgiving break: project proposal, user authentication implementation, and front-end – back-end contract/documentation. Additionally your group will be expected to meet with your TA once a week to share the progress each of you has made that week working on the project. This serves both to keep you on track and to give me a better idea of your individual contributions. After Thanksgiving break we will have three classes of group project presentations and demo sessions. You will be graded on each of these items separately, however, you will be given the opportunity to improve these submissions for reduced additional credit. Because of this, your group will not receive final grades on individual components until the project is complete. Additionally each member of your group may not receive the same grade. Individual contributions to the group project will be assessed on the basis of TA check-ins, peer-reviews, and (time permitting) an end of the semester meeting with the instructor.
2 Requirements

Your project **MUST** make appropriate use of the following technologies. This applies to all projects. If your project idea does not need to perform these functions, it is probably not appropriate as a project for this class.

- HTML and CSS
- Python-Flask
- Jinja2 Templates
- WTForms
- Flask-SQLAlchemy
- Flask-Login

In particular your application **MUST** do the following.

- Allow users to create accounts and store appropriately hashed passwords. This includes running your server over HTTPS.
- Accept user data and store it in your own database that you have designed and created.
- Provide that data in a meaningful way through a well formatted (and hopefully attractive) web interface.
- Serve a reasonable purpose. (You are not permitted to create a trivial application that nominally meets all requirements, but has no real purpose.)

3 Expectations

Projects that merely meet the minimum requirements will not receive A’s. The particular application chosen by your group likely invites the use of many additional technologies. I’ve listed below a number of examples. Some extra features may simply make more complex use of required technologies. If you are unsure what additional features might make sense for your application, please speak with me or your group TA. This is where you can be most creative with your project and make it your own.

- RESTful API and Ajax
- OAuth (client/server)
- Email confirmation on accounts for password reset
- Multi-factor Authentication
- Cloud Hosting
- Live Multi-User Interaction
4 Assignments

Over the remainder of the semester you will have a number of due dates and required check-ins. Each of these pieces is worth some portion of your project grade. The below table includes estimates of roughly how much each piece is worth.

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal</td>
<td>20%</td>
</tr>
<tr>
<td>User Authentication Check-In</td>
<td>20%</td>
</tr>
<tr>
<td>Front-End – Back-End Contract</td>
<td>20%</td>
</tr>
<tr>
<td>Class Presentation</td>
<td>10%</td>
</tr>
<tr>
<td>Private Demo/Final Submission</td>
<td>30%</td>
</tr>
</tbody>
</table>

4.1 Proposal – Due November 1st

You must submit a proposal for your project including the following. You are allowed, and even encouraged to meet with me or your group TA to discuss potential applications and features. Please note that the proposal is a required component of your project. Even if you miss the deadline, you are expected to hand in a proposal later for reduced credit. If the proposal is never handed in you cannot get an A on your project regardless of how impressive the final result. This is an important step in assessing the viability of your project and its fit with the course expectations.

- A description of your applications functionality and motivation – What will it do and why would someone want something that does that. (250 words)
- An outline of how your proposed project will meet the minimum project requirements.
- An outline of the ways your project will go beyond the minimum requirements.
- Assignments of who plans work on what parts of the application. (This is not necessarily final. Often you will find that some pieces are more or less difficult than you originally imagined.)
- The link to the GitHub repository you will be using as you build your application.
4.2 User Authentication – Due November 11th – 17th

This assignment will be handed in as a demo with your TA. You as a group are responsible for scheduling this meeting. Let me know if you are unable to work out a time to meet, however, if you wait to contact your TA until the last minute I am likely to be less understanding when you can’t find a time to meet. It is expected that after you have started running your web server, the TA will be permitted to use the browser on your machine to verify that it works as required.

Requirements

Your application must include the following pages for this check-in. To receive full credit it must also securely hash user passwords and save all user account information in a database. You are expected to use Flask-Login for this assignment. It is also expected that your code will handle common errors such as checking to see if a username is already in use during account creation.

- Application Home ("/") with links to create an account or log-in with an existing account
- Create Account – a form that takes username/password to be used to create a new account
- Login – a form that takes username/password and authenticates a user
- Hidden – a page that redirects anonymous users to the login page, but displays some basic information (such as username) for a user that is logged in