Group Project

Course: Web-Based Systems
Number: SENG 513
Semester: Winter 2024
Due Dates: See below
Instructor: Steve Sutcliffe <steve dot sutcliffe at ucalgary dot ca>
Version: 1.0.0

Objective:

This project aims to provide students with a comprehensive understanding of the principles and techniques involved in web-based system development, coupled with hands-on experience in building a mid-size web-based application within a collaborative software development team. The goal is to develop the necessary skills and knowledge to contribute to a web-based application successfully.

Overview:

This project will make up the 70% of your grade. This is a group project where you will build a mid-sized, responsive web-based application consisting of a back and front end. Groups should ideally be made of 5 people.

Through this project, you will cultivate the skills vital for a web app developer. These skills include interacting with a server and database via the command line and effectively demoing your application to potential investors, utilizing tools relevant to the industry. This project will equip you with practical, hands-on knowledge essential to web application development.

Groups will receive a project score weighted by peer evaluations (see the final submission section).

Project Selection

You can select the project, provided it meets the requirements outlined in the Project Requirements section.

Group Formation

You may form your groups freely. If you're forming groups or brainstorming project ideas, consider posting on the D2L forums to find teammates. Anyone not part of a group by the specified deadline below will be assigned to a group by the instructor or TAs. Additionally, teams with fewer than four members may merge with other groups or have individuals not yet in a group randomly assigned to them. This way, all teams will have an optimal size of approximately five members.

Project Evaluation

Given this project's comprehensive nature and practical importance, elements relevant to the project, such as the proposal, mock-ups, etc., will be evaluated more stringently compared to the other assignments. This rigorous evaluation reflects the field's high standards and encourages your best work, ensuring you gain the most from this experience. It is a chance to push your boundaries, showcase your abilities, and develop a strong foundation in web app development. I am confident that each of you will rise to the occasion and deliver work you're proud of, which will prepare you for the challenges and demands of a professional environment.

The Project will consist of the following parts:

Section	Grade %	Due Date (approx)
Group Formation Deadline*	0%	Jan. 22
Proposal	10%	Feb. 5
Repository Setup*	0%	Feb. 9
UI Mock-ups	10%	Feb. 16
Docker-Compose File*	0%	Mar. 1
Database Schema and Seeding Script*	0%	Mar. 15
Technical Topic Presentations Begin	10%	Mar. 19
Front-end Implementation*	0%	Mar. 28
Final submission and demos	40%	Apr. 9

*While these items do not have marks assigned, they will directly impact the final project mark. Some sections (e.g., Docker Compose File) will be evaluated before the final submission deadline. These have been noted in the marking guide below.

Project Requirements

Groups can select any web-based application they wish, provided it meets the following technical requirements:

Client-side

- Implements HTML, CSS, and JavaScript technologies.
- Must be accessible through a single-page interface.
- Must be fully compatible with Chrome.
- Must serve as a GUI frontend for the server component.

Mobile Support

- Must be fully responsive on desktop and mobile devices.
- Must provide a reasonable amount of functionality on a mobile device.
- Must demonstrate your strategy of either mobile-friendly or mobile-first design.
- Must handle unexpected disconnects and reconnects gracefully.

Server-side

- Majority of business logic must be performed on the server side.
- Communication between the server and client sides must be handled securely and effectively.

Multiple users & roles

- Needs to support five simultaneous users.
- Must implement user authentication.
- Must support at least **three** user roles (e.g., admins, guest, etc.).
- Must provide different functionality based on the user role.

Persistence

- Must demonstrate persistence capabilities, including server restarts, user disconnections/logout, etc.

Deployment

- Must be deployable in a packaged form using Docker.

Sample Projects

- Some of the previous project's demos are available on D2L.

Technologies to Consider

Your project should include at least one technology you have not used but would like to learn about. You can use almost any technology, provided it does not build your application through a WYSIWYG-type editor. E.g., technologies such as Angular, React, Vue.js, Express, Three.js, Tone.js, Bootstrap, RxJs, Webpack, MongoDB, MariaDB, Google Cloud, or serverless technologies such as AWS Lambdas or Firebase. Consider exploring technologies that make it easier for you to add features and demo your apps.

Proposal [60 Marks]

Your team must submit a single, well-structured PDF document with precise language to Dropbox on D2L. The document should be less than 10 pages (even 2-3 pages is fine) and include the sections below.

5-point and 10-point rubrics (see appendix) will be used for each section listed below unless otherwise indicated.

Team Members [3]:

List all members of your team [1], their student IDs [1], and the project title [1].

The Problem [10]:

Write two paragraphs stating the overall purpose of your proposed web-based application. In your first paragraph, you should provide some background and describe a problem. In your second paragraph, you should propose how to solve this problem.

The target audience for this part is a non-technical person, such as an investor deciding whether to give you money to develop your project.

User Requirements [5]:

Describe primary and secondary user groups/roles and their needs. For example, you may have players, administrators and hosts for an online game. Create and describe a personas for each role. Limit this to 2- 3 sentences. Give examples of 2-3 typical tasks/use cases that each persona would want to perform using your application and how they would accomplish them. This should be described at a very high level.

Functional and Implementation Requirements [10]:

This section aims to give some technical details about your proposed application. Feel free to use tables/lists in this section.

Identify and prioritize features you plan to implement in your project. Start by listing all the potential features your application could have. Then, categorize them based on their importance and value to the user experience. For example, consider a feature like 'settings' in your app, which allows users to update their personal information, such as email address or password. Evaluate how critical this feature is to the app's functionality and user experience. Is it essential for the initial release, or can it be developed later?

List all technologies you plan to use and what you want to use them for. Please note that what you write in this section is not binding. You can change technologies, priorities or even features as you start working on the project. For this part, we expect you to do some preliminary investigation into what technologies might work well with your project. For example, tools like Continuous Integration/Continuous Deployment (CI/CD) platforms have been instrumental in streamlining development processes and enhancing collaboration and utilities such as Ngrok provide support for local testing (i.e., broadcasting your project to a mobile phone).

This section will help us evaluate the scope of your project. We will be evaluating the ambition and feasibility of your project here, as well as the clarity and priority of the features and implementation requirements.

Mobile strategy [5]:

Describe and justify your mobile strategy, determining whether a mobile-first or desktop-first approach best suits different user groups and functionalities. For instance, if your app is centred on student registration, adopting a mobile-first design could enhance accessibility and user experience for students, ensuring the interface is intuitive and streamlined on smaller screens. Following this, you could then expand the design responsively for desktop use. Conversely, a desktop-first strategy may be more appropriate for administrators who require extensive functionalities and a more complex interface. In this scenario, it is crucial to deliberate on which features are essential for the mobile version, how they can be adapted to fit a smaller screen, or if certain functionalities will be exclusive to the desktop version. Your strategy should be well-rounded, ensuring both sets of users have a seamless experience while acknowledging the inherent trade-offs and adjustments required when transitioning between mobile and desktop versions.

Project Proposal Slide/PDF [10]:

Create a single-page presentation with images to showcase your project. Describe its concept, unique features, and functionality. Keep it clear and concise for the best engagement. It should have a reasonable representation of your app's intended appearance. You can think of this as a slide that would be used to get potential investors interested in looking at your work. Evaluation of this will consider both the slide's appearance and the content.

Project Timeline [5]:

Create a project timeline that you intend to follow. This timeline should serve as a roadmap for your team, outlining what tasks need to be completed and their respective due dates. When constructing your timeline, consider all the components and phases of your project. Start by breaking the project into manageable tasks and then assign realistic deadlines for each. For instance, early tasks might include completing UI Mock-ups. As you progress, include more detailed tasks, such as developing the base functionality of your application, followed by more advanced features. Be sure to include everything that needs to be done for the final submission. This timeline should help you keep your team on track and manage workload effectively throughout the duration of the project.

Definition of Roles and Responsibilities [10]:

Create a list of roles and responsibilities for the project and assign team members. When assigning these roles, consider the strengths and interests of your team members to maximize efficiency and engagement.

It is crucial that at least two people share each role. This approach facilitates collaboration and ensures continuity and support in case one team member is unavailable. For example, you might have roles like 'Database Management', which include database creation, maintenance, and optimization. In this case, assign at least two members to take charge of these tasks. Given the relatively small size of your groups, team members will likely need to take on multiple roles. This is a great opportunity to develop a range of skills and gain a comprehensive understanding of different aspects of the project and work with different members of your team.

A partial example of this follows on the next page.

Role	Responsibilities	Members
Database Design	Define database structure Normalization	Entire team
Database Implementation	SQL scripts Test data population Stored procedures Coordinate/Communicate	Darryl & Darlene
GUI Mockups	Gathers ideas from the entire team Lo-fi designs Iterates based on team feedback	Cathy & Carson

A partial example of the roles and responsibilities table.

SUBMISSION [2]:

Pick one member of your group who will submit two PDF files to D2L, one for the proposal and the other for the single-page slide:

- the proposal PDF should be named seng513-proposal-group-<N>.pdf (e.g., seng513-proposal-group-7.pdf)
- the slide PDF should be named seng513-slide-group-<N>.pdf (e.g., seng513-slide-group-7.pdf)

Repository Setup*

We will be using the University of Calgary GitLab server to host the repositories for this project. It is very straightforward to transition from one repo system to another, so if you want to put this project on GitHub as well (providing that is okay with your teammates), you can always do that at any time by simply adding another remote to your local repository.

To complete this section, you must:

- ☐ Create your repository on the university's GitLab server (https://csgit.ucalgary.ca/).
- ☐ Name that repository seng513-202401-group-<N> (e.g., seng513-202401-group-7).
- ☐ Share that repository with your instructor and both TAs.

It is expected that you will:

- ☐ follow convention standards for commit messages: https://www.conventionalcommits.org/en/v1.0.0/
- ☐ follow convention standards for branch naming: https://shorturl.at/ahlGZ
- ☐ avoid storing sensitive information such as passwords (passwords needed to run your project should be submitted on D2L in your final submission in config.yml files).

Once you have finished initializing your repo, create a text document with the Names and UCIDs of your entire team and the name of your repository as seng513-repository-group-<N>.txt to the Repository Setup Dropbox.

UI Mockups [25 marks]

As part of the project development process, you must submit a PDF containing mockups of your web application and brief descriptions of each page's functionality. These mockups should effectively capture the look and feel of your application, illustrating both the aesthetic design and user interactions.

Please ensure that your mockups demonstrate how users will navigate through the application. This includes indicating how different elements, like buttons or links, function, specifying that clicking a 'Profile' button will lead to the profile page.

Remember, the main goal of this milestone is to communicate the intended user experience and the flow of interaction within your app. It's understood that SOME DESIGN ASPECTS MAY EVOLVE AND CHANGE AS YOU PROGRESS IN YOUR DEVELOPMENT. The mockups guide your initial vision, providing a foundation FOR YOUR PROJECT TO grow and develop.

Overall Design and Aesthetics [4]:

4 - Exceptionally well-designed, visually appealing, and professional-looking UI.
3 - Good design with clear visual appeal; minor improvements needed.
2 - Satisfactory design but lacks visual appeal; noticeable areas need improvement.
1 - Poor design and not visually appealing; major improvements required.
0 - No effort in design; not visually appealing at all.

Consistency and Branding [4]:

4 - Highly consistent UI elements and strong, clear branding throughout.
3 - Generally consistent UI elements and good branding with minor inconsistencies.
2 - Some consistency in UI elements; branding is somewhat clear but needs improvement.
1 - Inconsistent UI elements and unclear branding; significant improvements required.
0 - No consistency in UI elements and no clear branding.

User-Friendly and Accessibility [4]:

4 - Exceptionally user-friendly design and highly accessible.
3 - Good level of user-friendliness and accessibility; minor improvements needed.
2 - Moderately user-friendly and accessible; noticeable areas need improvement.
1 - Not user-friendly or accessible; significant improvements required.
0 - Completely lacking in user-friendliness and accessibility.

Attention to Detail and Precision [4]:

4 - High attention to detail and precision in design elements.
3 - Good attention to detail and precision, but minor inaccuracies present.
2 - Satisfactory attention to detail, but noticeable inaccuracies and lack of precision.
1 - Poor attention to detail and precision; major inaccuracies present.
0 - No attention to detail or precision in design elements.

Innovativeness and Creativity [4]:

4 - Highly innovative and creative design.
3 - Good level of innovativeness and creativity; minor enhancements needed.

2 - Moderate level of innovativeness and creativity; noticeable enhancements needed.
1 - Lacks innovativeness and creativity; significant enhancements required.
0 - Completely lacking in innovativeness and creativity.

Mobile/Desktop Design Consideration [5]:

5 - Exceptional consideration for both mobile and desktop platforms; design is fully optimized for usability and aesthetics on both platforms.
4 - Good consideration for both mobile and desktop platforms; minor optimizations needed for perfect usability and aesthetics.
3 - Adequate consideration for both mobile and desktop platforms; noticeable areas of the design need optimization for better usability and aesthetics.
2 - Insufficient consideration for mobile and desktop platforms; significant areas of the design are not optimized, negatively affecting usability and aesthetics.
1 - No consideration for the differences between mobile and desktop platforms; the design is not usable or aesthetic on one or both platforms.
0 – Nothing submitted or designs are significantly lacking on both platforms

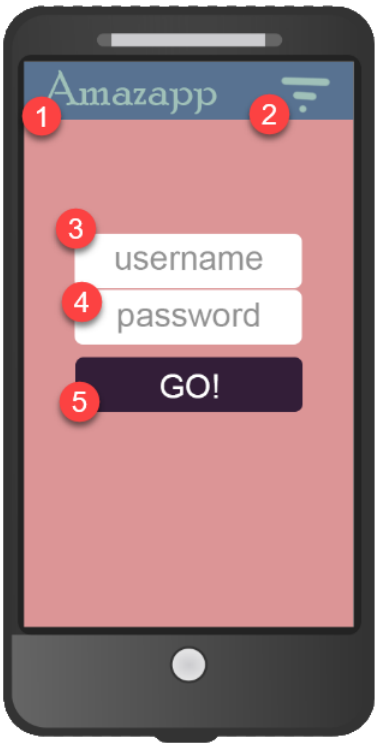
Submission:

Submit a single PDF of your mockups to Dropbox as seng513-mockups-group-<N> (e.g., seng513-mockups-group-7). While we encourage using Figma, it is not a requirement. If you used Figma, in addition to the pdf, include a link in a text document saved as seng513-figma-link-group-<N>.

An example is provided on the next page.

The provided mockup example below serves as a basic guideline to help you get started and understand the expectations for this project. However, please note that these examples represent a baseline passing grade, and your submissions should aim to surpass this standard. Pay close attention to detail, ensure thorough consideration of both mobile and desktop platforms, and strive for a higher level of polish and completeness in your work to achieve the best possible grade.

E.g. 1,

<table>
<tr>
<td rowspan="7"> 
</td>
<td>1. Logo: Poor Richard font, #A0C1B9
clicking the logo will bring the user to the description page</td>
</tr>
<tr>
<td>2. The hamburger menu selector will provide the options:
Register
Admin
About
Help</td>
</tr>
<tr>
<td>3. Text box for username, field is identified to the user by the grey placeholder text “username”</td>
</tr>
<tr>
<td>4. Text box for password, identified by the grey placeholder text “password”</td>
</tr>
<tr>
<td>5. Button: #331E38, checks if username/password are correct then takes the user to the main app page (see page 2).</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Phone image in public domain, from: https://publicdomainvectors.org/en/free-clipart/GSM-touch-screen-phone-vector-image/16936.html</td>
</tr>
</table>

Docker-compose file*

Docker Compose simplifies the process of managing multi-container Docker applications. Using a Docker Compose file, you can easily define and run multi-container Docker applications. It also simplifies setup for future developers or whenever you need to demo your project quickly.

You will need a good password strategy to manage your Docker setup since you do not want your passwords stored in your repository. Whatever strategy you use is up to you.

To complete this section, you must:

- ☐ Commit your docker-compose.yml and all other supporting Dockerfiles to your GitLab account.
- ☐ Create a repo tag at the commit point where your Docker configurations are ready for review. The tag should be named `project-docker-milestone`.
- ☐ Provide some information in a PDF document that includes the steps you need to set up your application and your password strategy. Name this file `seng513-deployment.pdf` and submit it to the Docker Compose Dropbox.
- ☐ Submit your docker-compose file and any other supporting setup files to the Docker Compose Dropbox in a zipped folder named seng513-deployment_files.zip.

The goal of this section should be for anyone coming over to your team to be able to set up and run your application quickly.

Your docker-compose and other setup documents may evolve over the remaining project time. That is to be expected. But this initial submission should demonstrate an easy-to-setup.

Database Schema and Seeding Script*

A database schema and seeding script are necessary components to get your application up and running quickly. You most likely would have this run with or immediately after your setup with Docker.

To complete this section, you must:

- ☐ Submit the schema script (e.g., the SQL script run that creates tables and defines relationships).
- ☐ Submit the seeding script (e.g.., the SQL script that populates tables with initial data).
- ☐ Instructions to run your schema and seeding scripts in a README.md.
- ☐ Commit these files to your repository and tag them as `project-schema-milestone`.
- ☐ Create a copy of your instructions and save them as seng513-schema-group-<N>.pdf.
- ☐ Compress your instructions and scripts into a *.zip and upload them to the Database Schema and Seeding Script Dropbox.

If your group is running a NoSQL database (like DynamoDB) or serverless architecture (like AWS Lambda with S3), then you should provide a document outlining the model (including the tables/keys/indexes/relationships) and include the schema-like representations (e.g., JSON structures, key-value pairs, etc.) and code snippets. There should be enough information there to let someone new to your team to understand and implement on a fresh set up of your project.

Technical Presentation [50 Marks]

In this technical presentation, each group will have the opportunity to showcase their project and delve into a specific technology they plan to utilize. This presentation is a chance to highlight your project's innovative aspects and demonstrate your understanding and mastery of the chosen technology. Technical presentations should be on a topic not discussed in class, tutorials, or by any other group presenting.

You will need to create and deliver a presentation to the class. The presentation should be about 12-13 minutes. At least three different people in your team should speak during the presentation. We will give 2 minutes after your presentation for the class to offer comments or ask questions. Each group has a strict 15-minute total time allotment, including the Q&A. It is critical to manage your time wisely, using the marks for each section as your guide. Presentations will be conducted on Zoom.

Project Overview [5]:

Introduce your group members and provide a **concise** overview of your project, outlining its purpose and critical functionalities. Include the single slide or pdf that you created to summarize your project. Using the following:

5 - Excellent: Demonstrates a comprehensive and clear understanding of the project, concisely summarizing the main objectives, scope, and significance. All group members are confidently introduced, and their roles are briefly explained. The overview is engaging and captures the audience's attention.
4 - Good: Provides a clear and coherent project summary, covering most of the key aspects and objectives. Group members are introduced, though some details may be missing. Engages the audience but may lack some clarity or enthusiasm.
3 - Satisfactory: Offers a basic summary of the project, but some key aspects or objectives are unclear or not covered. Group members are introduced, but details and roles may be vague. The presentation is understandable but may lack engagement.
2 - Needs Improvement: Provides an incomplete or unclear project summary, missing several key aspects or objectives. Group member introductions are minimal or unclear. The presentation struggles to engage the audience and lacks clarity.
1 - Poor: Fails to provide a coherent project summary, with most key aspects and objectives missing. Group members are not introduced, or the introduction is extremely unclear. The presentation does not engage the audience and is difficult to follow.

Introduction [5]:

Briefly introduce the technology and its relevance to your project. Optionally, provide a short background on the technology's development and evolution.

5 - Excellent: Provides a comprehensive and clear introduction to the technology, precisely explaining its relevance and significance to the project. Demonstrates an exceptional understanding of the technology and its application, engaging the audience and sparking interest.
4 - Good: Offers a clear introduction to the technology, adequately covering its relevance and significance to the project. Shows a good understanding of the technology, but minor details may be missing. Engages the audience effectively.
3 - Satisfactory: Provides a basic introduction to the technology, but some aspects of its relevance or significance to the project may be unclear. Shows an adequate understanding of the technology, though further detail could enhance clarity. Engagement with the audience is present but could be improved.
2 - Needs Improvement: Gives an incomplete or unclear introduction to the technology, with many aspects of its relevance or significance to the project left unexplained. Shows a limited understanding of the technology. Struggles to engage the audience and lacks clarity.
1 - Poor: Fails to provide a coherent introduction to the technology, leaving out critical information about its relevance or significance to the project. Demonstrates a lack of understanding of the technology. Does not engage the audience and is difficult to follow.

Why is the Technology Helpful [5]:

Explain how this technology enhances your project and contributes to achieving your goals.

5 - Excellent: Provides a compelling and detailed explanation of why the technology benefits the project, supported by specific examples and clear reasoning. Demonstrates an in-depth understanding of the technology's impact, engaging the audience and articulating the value effectively.
4 - Good: Offers a clear and convincing explanation of the technology's benefits, with relevant examples and sound reasoning. Demonstrates a good understanding of its positive impact on the project, maintaining audience interest.
3 - Satisfactory: Provides a basic explanation of why the technology is helpful, but some details or examples may be lacking. Shows an adequate understanding of its benefits, though the connection to the project could be more robust. Engagement with the audience is present but could be improved.
2 - Needs Improvement: Gives a vague or incomplete explanation of the technology's benefits, with insufficient examples or unclear reasoning. Demonstrates a limited understanding of its positive impact on the project. Struggles to maintain audience engagement and lacks clarity.
1 - Poor: Fails to explain why the technology is beneficial, lacking specific examples and clear reasoning. Demonstrates a lack of understanding of its impact on the project. Does not engage the audience and is difficult to follow.

Life Without the Technology [5]:

Discuss the challenges and limitations your project would face without this technology.

5 - Excellent: Provides a vivid and detailed portrayal of the challenges and limitations that would be faced without the technology, making a strong case for its necessity. Demonstrates a deep understanding of the technology's impact, capturing the audience's attention and emphasizing the contrast effectively.
4 - Good: Offers a clear and well-reasoned explanation of the difficulties that would arise in the absence of the technology, showcasing its importance. Demonstrates a good understanding and creates a noticeable contrast, maintaining audience interest.
3 - Satisfactory: Gives a basic description of the potential challenges without the technology, but some details may be lacking or unclear. Demonstrates an adequate understanding, though the contrast could be highlighted better. Engages the audience but could be more impactful.
2 - Needs Improvement: Provides a vague or incomplete depiction of life without the technology, lacking specific details and clear reasoning. Demonstrates a limited understanding and fails to create a strong contrast. Struggles to maintain audience engagement.
1 - Poor: Fails to explain the challenges coherently without the technology, lacking detail and clarity. Demonstrates little to no understanding of its importance, failing to engage the audience and create a contrast.

Description of the Technology [10]:

Offer a detailed explanation of how the technology works and its key features.

10 - Exceptional: Provides an exceptionally clear, comprehensive, and detailed description of the technology, covering all essential aspects and functionalities. Demonstrates an in-depth understanding and mastery of the subject, captivating the audience's attention.
9 - Outstanding: Delivers an outstanding description, addressing nearly all key aspects and functionalities of the technology with great clarity and depth. Maintains strong audience engagement and demonstrates a thorough understanding.
8 - Very Good: Provides a very good description, clearly covering most of the important aspects and functionalities. Shows a solid understanding of the technology, keeping the audience engaged.
7 - Good: Offers a clear and adequate description but may miss a few minor aspects or details. Demonstrates a good understanding, with some room for enhancement in-depth and audience engagement.
6 - Above Average: Gives a satisfactory description, covering the basic aspects and functionalities, but lacks some detail and depth. Demonstrates an above-average understanding, with noticeable areas for improvement in clarity and engagement.
5 - Average: Provides an average description, missing several key aspects or details and lacking in depth. Shows a basic understanding, but struggles to maintain consistent audience engagement.
4 - Below Average: Delivers a below-average description, missing significant aspects and functionalities, and lacking clarity. Demonstrates a limited understanding, with weak audience engagement.
3 - Needs Significant Improvement: Offers a vague and incomplete description, failing to address critical aspects and functionalities. Shows minimal understanding and struggles to engage the audience.

2 - Poor: Provides an unclear and inadequate description, with an apparent lack of understanding of the technology. Fails to engage the audience and lacks coherence.
1 - Extremely Poor: Fails to provide any meaningful description of the technology, demonstrating no understanding of the subject. Completely fails to engage the audience, lacking any form of clarity or coherence.

Quick Tutorial and Live Demo [10]:

Provide a hands-on demonstration of how to use the technology, showcasing its application in your project.

10 - Exceptional: Delivers an exceptionally clear, well-structured, and engaging tutorial, complemented by a flawless live demo that effectively demonstrates the technology's functionality. Exhibits mastery in guiding the audience through each step, ensuring comprehensive understanding.
9 - Outstanding: Provides an outstanding tutorial with a highly effective live demo, showcasing the technology's capabilities with minimal issues. Demonstrates excellent instructional skills, maintaining strong audience engagement.
8 - Very Good: Offers a very good tutorial and live demo, covering most functionalities with clarity. Any issues are quickly resolved, showing good preparation and a solid understanding of the technology.
7 - Good: Gives a clear tutorial and live demo, but may miss some minor functionalities or details. Demonstrates a good level of understanding and maintains audience interest, despite minor hiccups.
6 - Above Average: Conducts an above-average tutorial and live demo, covering the basic functionalities but lacking some detail and depth. Shows an above-average understanding, with noticeable areas for improvement in execution and engagement.
5 - Average: Provides an average tutorial and live demo, missing several key functionalities and details. Demonstrates a basic understanding but struggles to maintain consistent audience engagement.
4 - Below Average: Delivers a below-average tutorial and live demo, with significant gaps in coverage and clarity. Demonstrates a limited understanding, with weak audience engagement and noticeable issues in execution.
3 - Needs Significant Improvement: Offers a vague and incomplete tutorial, with a live demo that fails to showcase the technology adequately. Shows minimal understanding, struggling significantly to engage the audience.
2 - Poor: Provides an unclear and inadequate tutorial, with a live demo demonstrating a lack of preparation and understanding of the technology. Fails to engage the audience and lacks coherence.
1 - Extremely Poor: Fails to provide any meaningful tutorial or live demo, demonstrating no understanding of the subject. Completely fails to engage the audience, lacking clarity, coherence, or preparation.

Competing Technologies [5]:

Discuss alternative technologies, comparing their pros and cons to your chosen technology.

5 - Excellent: Provides a comprehensive and well-articulated comparison of the chosen technology with its competitors, highlighting key differences, strengths, and weaknesses. Demonstrates an in-depth understanding and presents the information engagingly and clearly.
4 - Good: Offers a clear and thorough comparison, covering most of the important aspects of the competing technologies. Shows a good understanding and maintains audience interest, though minor omissions or lack of detail might exist.
3 - Satisfactory: Gives a basic comparison of the chosen technology with its competitors, but some key aspects might be missing or unclear. Demonstrates an adequate understanding, but there is room for improvement in clarity and engagement.
2 - Needs Improvement: Provides a vague or incomplete comparison with significant gaps in coverage and clarity. Demonstrates a limited understanding of the competing technologies and struggles to maintain audience engagement.
1 - Poor: Fails to provide any meaningful comparison or information about the competing technologies, showing a lack of understanding and preparation. Does not engage the audience and lacks coherence and clarity.

Delivery and Questions [5]:

The presentation should last approximately 12-13 minutes. We'll allow for 2 minutes of questions at the end. At least half of your team should take turns presenting material.

5 - Excellent: All or most team members actively participate, keeping the presentation within the 8-10 minute timeframe and answering questions with clarity and confidence. The presentation is well-paced and engages the audience throughout.

4 - Good: At least half of the team members participate, and the presentation stays mainly within the 8-10 minute timeframe. The team answers questions effectively, maintaining a good level of audience engagement.
3 - Satisfactory: Just under half of the team members actively participate, causing slight deviations from the 8-10 minute timeframe. The pacing is adequate, but there may be moments of rushing or dragging. Answers to questions are generally clear, though improvements could be made.
2 - Needs Improvement: Participation from the team is limited, and the presentation struggles to stay within the 8-10 minute timeframe. There are noticeable pacing issues, and the responses to questions lack clarity, affecting audience engagement.
1 - Poor: Very few or no additional team members participate, resulting in a presentation that exceeds or falls short of the 8-10 minute timeframe. Answers to questions are unclear, leading to poor audience engagement and confusion.

Note: If less than half of the team members participate, regardless of the other criteria, the score for this section should be reduced to reflect the lack of participation.

SUBMISSION

There is no submission required for this section.

Front-end/UI Implementation*

This phase is crucial for laying down your web application's visual and structural groundwork.

To complete this section, you must:

- Have a static version of your web application running, including a responsive design with external CSS.
- Have UI elements and layout implemented based on your design mock-ups (with revisions).
- Tag your commit with this stage complete as `project-milestone-front-end`.
- Include a *.pdf that comments on how your implementation has changed from your mock-ups and why. Submit this as seng513-ui-group-<N>.pdf to the Front-End Implementation Dropbox.
- Include a *.pdf of screenshots of all pages of your current implementation. Save this as seng513-ui-implementation-group-<N>.pdf to the Front-End Implementation Dropbox.

Final Submission and Demos [125 Marks]

You will need to submit your code via a repository. Be sure to add your instructor and both TAs to your repository.

You will need to provide two video demos; one should be a product showcase and illustrate your functionality, and the other should be a technical demo showcasing the technologies you used, your implementation, and anything else you want to brag about at a potential interview.

Please note that the product showcase video may be shared with future students as an example of exemplary work. Should you wish to keep your project presentation confidential and not have it used for educational purposes beyond the current scope, kindly display the following message at the beginning of your video: "Confidential: Not for External Display." This will indicate that your video is intended for internal grading and feedback only within this course term.

1. Peer Evaluations

Peer evaluations will affect your score on the project. When conducting peer evaluations, be sure you evaluate the work of the others on the project and not the individuals themselves. A peer evaluation Excel sheet is provided to you in D2L. Submit these to Peer Evaluations Dropbox as seng513-peer-evals.xlsx. This is optional; any evaluations not submitted will assume a score of 5.

Using the 5-point rubric in the appendix of this document, evaluate your peers using the following sections:

Evaluation	Description	Score (out of 5)
Contribution to the project	Does the team member consistently and reliably contribute to the project? Do they complete all tasks on time without reminders?	
Quality of work	Is the code well-organized, and does it follow best practices? Was the code easy to integrate into the project?	
Communication skills	Does the team member communicate well with the team and regularly share updates?	
Teamwork/Collaboration	Is the team member generally supportive, working well with others, and showing respect to all other members?	
Problem-Solving Skills	How well does this team member address and resolve issues as they arise?	
Adaptability/Openness to Feedback	Does this team member actively seek feedback from other team members and make changes accordingly?	

Final project scores will be adjusted according to these scores using a weighted adjustment factor of 20% of the final project grade.

For example, in a team of 5, where the group scored 82% on the project grade:

Group project score: 82%
Darlene's Peer Evaluation Score: 93/120 (6 evaluation sections * 5 points * 4 team members = 120 total points)
Darlene's Adjustment Factor: 93/120 = 0.775
Peer evaluation deficit: 1 – 0.775 = 0.225 (she was short 0.225 of a perfect score)
Weighted adjustment (20%): 1 – (0.2 x 0.225) = 0.955
Adjusted project score: 82% * 0.955 = 78.31%
Darlene would receive a 78% => B

*If a team member receives a score of 0 on their peer evaluation, indicating a complete lack of participation or contribution, a separate evaluation may be conducted to assess their involvement in the project. The instructor will carry out this evaluation, which could include a personal interview, a review of individual contributions, and an analysis of communication within the team. This evaluation will determine whether additional penalties to the team member's grade are warranted, ranging from a reduced project grade to a failing grade for the entire course, depending on the severity of non-participation.

2. Marking Guide for Final Group Project

2.1 Contribution and Programming Roles [10]

The frequency and quality of commits, branch naming, and commit messages are evaluated. The final release is tagged as "Release 1.0". This section is based on your Git repository, including the initial setup. The following parts are evaluated:

[2] Initial setup evaluation (name correctly: seng513-202401-group-<N>, shared, delivered on time) as well as the other relevant milestones (e.g., project-docker-milestone).

[2] The final release is tagged as "Release 1.0".

[2] Commit frequency and quality (follows standards).

[2] Branches are used effectively and follow standard conventions.

[2] Sufficient code is written by all members.

2.2 Product Showcase Video [30]

A brief video (approx. 5 minutes) that summarizes the problem and its importance and demonstrates your product's solution to the problem. This is meant to be shown to potential clients and investors. Avoid technical jargon; you don't need to reveal anything "behind the curtain." Instead, this should demonstrate what your product does. This is fantastic to have in your portfolio to show future employers.

Problem Statement Clarity [5]

5: Exceptionally clear and concise presentation of the problem. The importance is compellingly articulated.
4: Clearly describe the problem with a solid explanation of its importance.
3: Adequate definition of the problem, but the explanation of its importance is somewhat vague or not fully compelling.
2: The problem is mentioned, but important details are missing or unclear.
1: The problem is poorly defined or missing entirely; the importance is not convincingly explained.

Professionalism and Production Quality [5]

5: High-quality video with professional visuals, sound, and editing. Captivating and appropriate for external stakeholders.
4: Good production quality with minor flaws in visuals or sound. Generally professional.
3: Satisfactory production quality, but with noticeable issues in visuals, sound, or editing.
2: The video has significant issues that detract from its professionalism.
1: Many video elements are unprofessional; it is unsuitable for stakeholders without major revisions; too much technical jargon.

GUI Showcased [10]

10: Exceptional display of UI design - visually appealing, user-friendly, and demonstrating outstanding attention to detail.
8: Very good UI design - aesthetically pleasing and functional, with minor areas for improvement. Demonstrates a good understanding of user experience principles.
6: Satisfactory UI design - generally presentable with a functional layout, but lacks refinement and has some usability issues.
4: Below-average UI design - noticeable issues in layout and aesthetics that impact user experience.
2: Poor UI design - unappealing, difficult to navigate, and significantly detracts from the user experience.

Functionality Showcased [5]

5: The functionality is demonstrated excellently, showcasing a fully working application with all features performing as intended. Highlights complex or innovative functionalities effectively.
4: Good demonstration of functionality - most features work as intended, but there might be minor glitches or unimplemented aspects.

3: Satisfactory functionality - the basic features are demonstrated, but noticeable issues or limitations exist in what is showcased.
2: Limited functionality - significant issues or shortcomings in the application's features, impacting the overall demonstration.
1: Poor functionality - critical features are missing, non-functional, or have serious issues, making the application largely unusable in its current state.

Solution Effectiveness [5]

5: The solution is demonstrated brilliantly, with clear, tangible benefits directly addressing the problem.
4: The solution is well-presented and shows how it addresses the problem effectively.
3: The solution is shown, but the connection to the problem or its effectiveness is not convincingly presented.
2: The solution relates to the problem but does not clearly show how it is solved.
1: The solution is demonstrated, but how it addresses the problem is unclear.

Additional Notes:

Avoid technical jargon and focus on the product's functionality
Captivate potential clients and investors, highlighting the product's value proposition.
Remember that this would be used in your professional portfolio; keep the video quality excellent.

2.3 Technical Demo Video [50]

The technical demo video is your opportunity to showcase the intricate workings of your project to a technically astute audience. Imagine you are in a job interview with senior developers and technical leads, explaining your role in a complex software project. Your demo should highlight the advanced features of your application, elucidate the technologies employed, and delve into significant segments of your code. It must convincingly demonstrate the following:

- Full Functionality of the App
- Mobile Support
- Multiple-User Support and Roles
- Data Persistence

This demo isn't just a tour of your project; it's proof of your technical prowess and your team's ability to solve real-world problems with cutting-edge solutions. This technical demo will play a part in evaluating your final project.

There is no video-length requirement or restriction for the technical demo; however, it should be long enough to cover all elements adequately but concise enough to maintain engagement. Most demonstration videos would probably be 7 to 10 minutes, but this is merely a guideline. Remember, the goal of this video is to confirm the presence and integration of the critical features of the project.

The technical demo video will be vital in marking the sections below. Please include a list of Please include a video timestamp index with your video submission (seng513-timestamps-group-<N>.txt). This index should list key timestamps corresponding to the various sections or topics covered in your video. For example, if you start discussing the mobile at 02:15 in the video, you would note this in your index. This makes navigating to specific parts of your presentation more accessible for the reviewers.

Complete Functionality of the App [10]

In your video presentation, showcase all the features of your app, particularly those you outlined in your initial proposal. Also, if your team went beyond the original scope and included extra features, be sure to include these. Evaluation will be conducted using the 10-point rubric in the appendix.

Coverage of Technical Requirements [10]

Demonstrate a thorough understanding of the technical requirements. This is the time to show off to a potential employer and brag about some of the code you are particularly proud of. Evaluation will be conducted using the 10-point rubric in the appendix.

Mobile Support [10]

Evaluation of the responsiveness on desktop and mobile devices, reviewing the mobile strategy, including mobile-friendly or mobile-first design. As well as test the handling of unexpected disconnects/reconnects. Using the 10-point rubric in the appendix.

Multi-User Support and Roles [10]

Multiple users are supported (at least five simultaneous users). Implement user authentication and at least three user roles with appropriate functionalities (different functionality for different users). Using the 10-point rubric in the appendix.

Data Persistence [10]

Persistence capabilities have been demonstrated through various user and server scenarios, e.g., the server is restarted (shut down and start-up). Using the 10-point rubric in the appendix.

2.4 Deployment [10]

The project can be deployed in a packaged form using Docker or similar technologies. A setup guide is included in the README in your repository. Evaluation is based on the Docker Compose File and Database Schema & Seeding Scripts sections submitted earlier. Using the 10-point rubric in the appendix.

2.5 Reflection [20]

Reflect on the development process, design evolution, knowledge gaps, and the overall project outcome. This reflection should offer insights into your experiences, what you have learned, and how you have addressed challenges throughout the project lifecycle. This is the most valuable section for any future employer. Answer the following questions in a “reflection.pdf”:

Schedule:

Discuss any deviations from your initial project timeline, explain what factors led to these changes, and how you adapted your schedule.

GUI Design Evolution:

Compare your original GUI design with the final implementation and highlight the changes and the reasons behind these modifications.

Unexpected Challenges:

Identify what aspects of the project were unexpected or not planned for, and reflect on how these challenges impacted the project and how you overcame them.

Knowledge Gaps:

Reflect on what knowledge or skills you wish you had before you started the project and discuss how you addressed these gaps during the course of the project.

Successes and Shortcomings:

Evaluate what worked well in the project and what did not and consider the factors contributing to the successes and the difficulties. Be sure to include whether your release plan goals were met, and if not, why not.

Overall Project Rating:

Rate your project on a scale of 1 (poor) to 10 (fantastic). Provide a rationale for your rating, considering the quality of the final product, the process of working on the project, and the learning outcomes. Note a higher evaluation would require a significant answer.

Reflection Grading:

This section will evaluate relevance/completeness [10] and constructive self-evaluation [10] using the 10-point rubrics in the appendix.

2.6 Submissions [5]:

Video Showcase:

Videos should be in the *.mp4 format. If you do not have access to specialized software, you can record a video using Zoom, which will record in the *.mp4 format. Ensure your video plays with VLC.

Upload your showcase video to the Showcase Video Dropbox. The video should be named seng513-showcase-group-<N>.mp4.

Technical Demo

Videos should be in the *.mp4 format. If you do not have access to specialized software, you can record a video using Zoom, which will record in the *.mp4 format. Ensure your video plays with VLC.

Upload your technical video to the Technical Demo Dropbox. The video should be named seng513-technical-demo-group-<N>.mp4. Include the video timestamp index as seng513-timestamps-group-<N>.txt.

If, for some reason, your video file size is too large for D2L, then upload your video to https://yuja.ucalgary.ca/ and share it with your instructor and both TAs. If you need to upload your video to Yuja, submit a ReadMe.txt document to Dropbox with your team names, UCID, and the video link.

Reflection

Upload your reflection as a PDF to the Final Project Submission Dropbox as seng513-reflection-group-<N>.pdf.

Appendix

Rubrics Used

Understanding Rubric-Based Grading

As you progress through your project and receive feedback based on our rubrics, it's essential to have the right perspective on what these scores represent. Rubrics are designed not just to evaluate your work but also to recognize and acknowledge the quality of your efforts. When you see a score like 4/5 or 7/10, rather than viewing this as having lost 1 or 3 marks, it's more constructive to see it as an affirmation that you have done very well.

Here's how to interpret rubric scores:

Scores such as 4/5 or 7/10: These are strong scores, indicating that your work is good and meets the majority of the criteria at a high level. It shows that you have grasped the project requirements well and effectively applied your knowledge.

Perfect Scores: While achieving a perfect score is excellent, it's important to remember that rubrics are also a tool to identify exceptional or outstanding work. Not every good or great project will hit every mark for a perfect score, and that's okay.

Purpose of Rubrics: The aim of using rubrics is not solely to pinpoint errors or shortcomings. Instead, they are a means to highlight areas of strength and identify when work is exceptional or goes above and beyond expectations.

Growth and Learning: Use the rubric as a guide for areas of improvement and professional growth. It's an opportunity to understand what can be enhanced in your future projects.

We encourage you to view the rubrics as a positive tool for feedback and growth. They are here to help you recognize the quality of what you've accomplished and provide guidance on achieving excellence in your work.

This approach to grading with rubrics is similar to receiving a letter grade for your work. Like how a letter grade summarizes your overall performance in a course or assignment, rubrics help break down your project into 'evaluation chunks.' This breakdown allows you to see more detail in the evaluation process.

By segmenting the project into distinct categories, rubrics offer specific feedback on each aspect of your work. This detailed approach not only makes it more evident where your strengths lie but also pinpoints the specific areas where there's room for improvement. It's about understanding the nuances of your performance in each part of the project rather than just looking at an overall grade.

Think of it as getting a more comprehensive picture of your work, which is instrumental in guiding your learning journey and helping you hone your skills in a targeted manner

5-Point Rubric

Unless otherwise stated, the following rubric will be used for 5-point assessments

5 - Excellent

- Fully meets and exceeds the expectations for the task or section.
- Demonstrates thorough understanding and clarity.
- For code: Efficient, effective, and free of errors.
- For written sections: Cohesive, comprehensive, and concise with zero to minimal errors.

4 - Good

- Meets most of the expectations for the task or section.
- Demonstrates good understanding with minor lapses.
- For code: Mostly efficient and effective with minor room for improvement or few errors.
- For written sections: Mostly cohesive and comprehensive with minor gaps or errors.

3 - Satisfactory

- Meets the basic expectations for the task or section but lacks depth.
- Demonstrates an adequate understanding.
- For code: Functional but may have inefficiencies or areas of improvement.
- For written sections: Adequate in addressing the topic but may lack clarity or have several errors.

2 - Needs Improvement

- Partially meets the expectations for the task or section.
- Demonstrates limited understanding.
- For code: Contains several errors or inefficiencies, requiring revision.
- For written sections: Limited in addressing the topic, may have numerous errors or gaps.

1 - Unsatisfactory

- Does not meet the expectations for the task or section.
- Demonstrates little to no understanding.
- For code: Non-functional or riddled with errors.
- For written sections: Largely off-topic, incoherent, or riddled with errors.

10-Point Rubric

9-10 - Excellent

- The section is exceptionally well-organized. Ideas are clear, concise, and effectively communicated.
- All information and code are highly relevant, and accurate, and show a deep understanding of the subject matter.
- The content is complete, addressing all required aspects thoroughly and leaving no gaps in information or functionality.
- Demonstrates outstanding innovation and creativity, providing unique and effective solutions or perspectives.
- Exhibits a high level of technical proficiency and mastery in both written and code sections, following best practices and showing attention to detail.

7-8 - Good

- The section is well-organized and ideas are mostly clear, though there may be minor issues with clarity or organization.
- Most of the information and code are relevant and accurate, but there may be minor inaccuracies or areas lacking depth.
- The content is largely complete, but there may be minor gaps in information or functionality.
- Shows a good level of innovation and creativity, though some aspects may be conventional or lack originality.
- Demonstrates a solid level of technical proficiency in both written and code sections, with only minor issues.

5-6 - Satisfactory

- The section is somewhat organized, but there are issues with clarity or structure that hinder understanding.
- Some information or code may be irrelevant or inaccurate, and there are noticeable gaps in knowledge or functionality.
- The content is somewhat complete, but there are notable gaps in information or functionality.
- Shows a basic level of innovation and creativity, but largely relies on conventional methods or ideas.
- Demonstrates an adequate level of technical proficiency in both written and code sections, but there are noticeable issues.

3-4 - Needs Improvement

- The section is poorly organized, and ideas are unclear or confusing.
- A significant portion of the information or code is irrelevant, inaccurate, or shows a lack of understanding.
- The content is incomplete, with significant gaps in information or functionality.
- Lacks innovation and creativity, with little to no original thought evident.
- Demonstrates a below-average level of technical proficiency in both written and code sections, with significant issues.

1-2 - Unsatisfactory

- The section is disorganized and ideas are not communicated effectively.
- The majority of the information or code is irrelevant, inaccurate, or demonstrates a lack of understanding.
- The content is highly incomplete, with critical gaps in information or functionality.
- Shows no innovation or creativity, and does not provide effective solutions or perspectives.
- Demonstrates a lack of technical proficiency in both written and code sections, failing to follow best practices.

0 - Incomplete or Missing

- The section is missing, or the content provided is incomplete to the point that no assessment can be made.