Assignment 4: Research, Plan, Design and Build a Single-Page Web Application – 40%

GRIFFITH COLLEGE DUBLIN

Student name:	**3046155**					
Student number:						
Faculty:	**Computing Science**					
Course:	**BSCH/BSCO**			**Stage/year:**	**3**	
Subject:	**Web Technologies**					
Study Mode:	Full time	☑		***Part-time***		
Lecturer Name:	**Gemma Deery**					
Assignment Title:	**Assignment 4: Research, Plan, Design and Build a Single-Page Web Application (node.js and angular)**					
Date due:	**21/03/2023**					
Date submitted:	**14/05/2023**					

Plagiarism disclaimer:

I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.

I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.

Signed: _____ **Date:** _____

Please note: Students MUST retain a hard / soft copy of ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

SINGLE PAGE APP

1 Contents

2 Introduction 3
2.1 Description of the project and purpose of document (1-2 paragraphs)Front-end functionality 3
2.2 Common features 3
2.3 Home page 4
3 Back-end functionality 4
3.1 How to log in and make changes 4
3.2 User Groups/ Permissions 4
3.3 Post Question 5
3.4 Answer Question 7
3.5 Filter out questions 7
3.6 Approve answers 7
3.7 Rate answers 8
3.8 Upvote 8
3.9 List answers 8
3.10 Search 8
3.11 Create comment 9
4 Design specifications 9
4.1 Inspired by 9
4.2 Sitemap 9
4.3 Wireframe templates 10
4.4 Colour 10
4.5 Fonts 11
4.6 Photo or image treatment 11
4.7 General style 11
5 Usage examples 12
6 Application Structure 12
7 Database Structure 13
8 The project timeline 13
9 Reflection on how the project went 13

2 Introduction

2.1 Description of the project and purpose of document (1-2 paragraphs)Front-end functionality

The project involves the development of a single-page web application designed to cater to curious trying to learn and seeking expert advice. Users can sign up and log in to post their questions, while experts can provide answers. The application allows users approve answers and give ratings. It also features an upvoting system to highlight the best answers and a search function to prevent duplicate questions. Users can engage in discussions by commenting on expert-submitted answers. The front-end functionality will be implemented using HTML, CSS, and JavaScript, with Angular as the web application framework. The aim is to create a user-friendly and intuitive interface that facilitates effective knowledge sharing and problem-solving within the community.

The purpose of this document is to provide an overview of the project's objectives and requirements. It outlines the features, architecture, wireframes, and database structure of the web application. The document also includes a project timeline and reflections on the project's progress. By building this application, the project aims to offer a platform where users can seek expert advice, share insights, and find valuable solutions, ultimately benefiting the users.

2.2 Common features

1. User Registration and Login: The application allows users to sign up for an account using their credentials and provides a secure login system for authenticated access to the platform. This feature ensures that users can create and manage their profiles effectively.
2. Question Posting: Users can post questions related to their home improvement or DIY problems. This feature enables users to seek advice and solutions from experts and the community.
3. Answering Questions: Experts can provide answers to the posted questions. This feature allows knowledgeable individuals to share their expertise and assist users in solving their problems.
4. Answer Approval: Users can approve answers that they find helpful or satisfactory. This feature allows the community to validate the quality of the provided solutions.
5. Upvoting: Users can upvote the best answers they come across, indicating their agreement with the solution's quality. This feature helps highlight the most valuable answers and provides a ranking system within the platform.
6. Search Functionality: The web application includes a search function to prevent users from posting duplicate questions. Users can search for existing questions and answers to find relevant information before posting a new query.
7. Commenting System: Users can engage in discussions and share additional insights by commenting on expert-submitted answers. This feature promotes knowledge sharing, collaborative problem-solving, and community interaction.

2.3 Home page

- Navigation Menu: A navigation menu is included to help users navigate through different sections of the application. It typically contains links to essential pages like Home, Questions, Login/Sign Up, and About.
- Search Bar: The home page may feature a search bar that allows users to quickly search for existing questions and answers before posting new queries. This helps prevent duplication and encourages users to find relevant information efficiently.
- Featured Questions: The home page may showcase a selection of featured or popular questions to grab users' attention and entice them to explore further.

3 Back-end functionality

- Server-Side Framework: The application utilizes Node.js as the server-side framework. Node.js provides a scalable and efficient runtime environment for executing server-side JavaScript code.
- User Authentication: The back-end handle's user authentication, allowing users to securely register, log in, and manage their accounts. This includes verifying user credentials, managing sessions, and enforcing access control to protect user data.
- Database Management: Due to some difficulties in implementing the database, the app uses in-memory objects so when you refresh the app the data will be lost.
- Data Validation and Sanitization: The back-end implement's robust validation and sanitization mechanisms to ensure the integrity and security of user input. This includes validating form data, sanitizing user-generated content to prevent injection attacks, and enforcing data integrity constraints.

3.1 How to log in and make changes

Admin email: admin@gmail.com

Password: root

3.2 User Groups/ Permissions

The single-page web application supports three types of users with varying permissions and capabilities:

1. Expert User: The expert user can ask questions and provide answers. They can utilize their knowledge and expertise to contribute to the community by responding to queries and sharing valuable insights.

2. Admin User: The admin user possesses enhanced privileges and responsibilities. In addition to asking questions and providing answers like the expert user, they have additional administrative

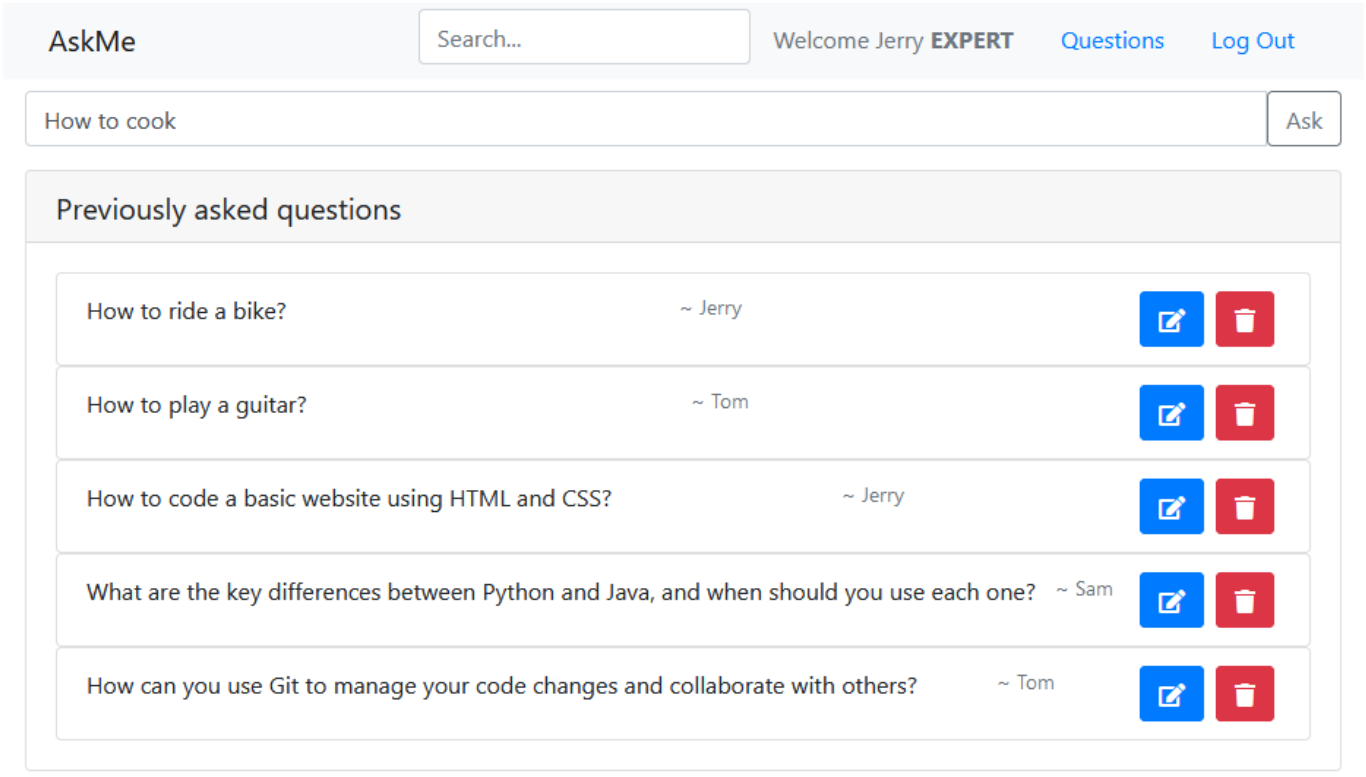
powers. These powers include the ability to edit questions, delete questions, and delete answers. Admin users play a crucial role in maintaining the integrity and quality of the platform's content.

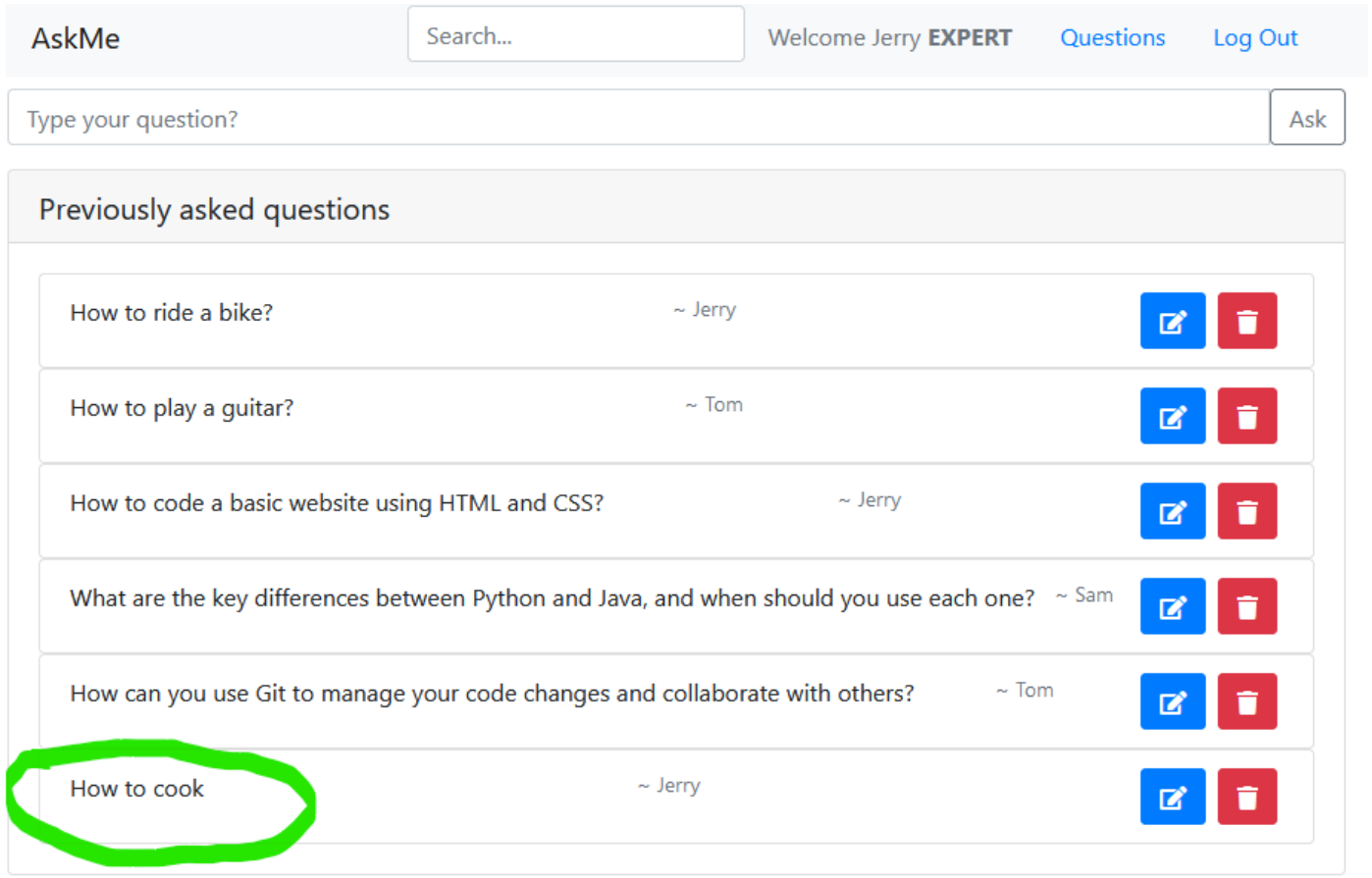
3. Normal User: The normal user has more limited capabilities compared to the expert and admin users. They can ask questions to seek guidance and solutions from the community. Additionally, they can engage in discussions by commenting on questions and answers. While normal users cannot edit or delete questions, their participation through comments and questions contributes to the collaborative and interactive nature of the platform.

These user types and their respective permissions help ensure that the single-page web application caters to a diverse range of users, including those seeking assistance, those providing expert advice, and those administering the platform. The distinct roles and capabilities of each user type enhance the functionality and effectiveness of the application in facilitating knowledge sharing and problem-solving within the community.

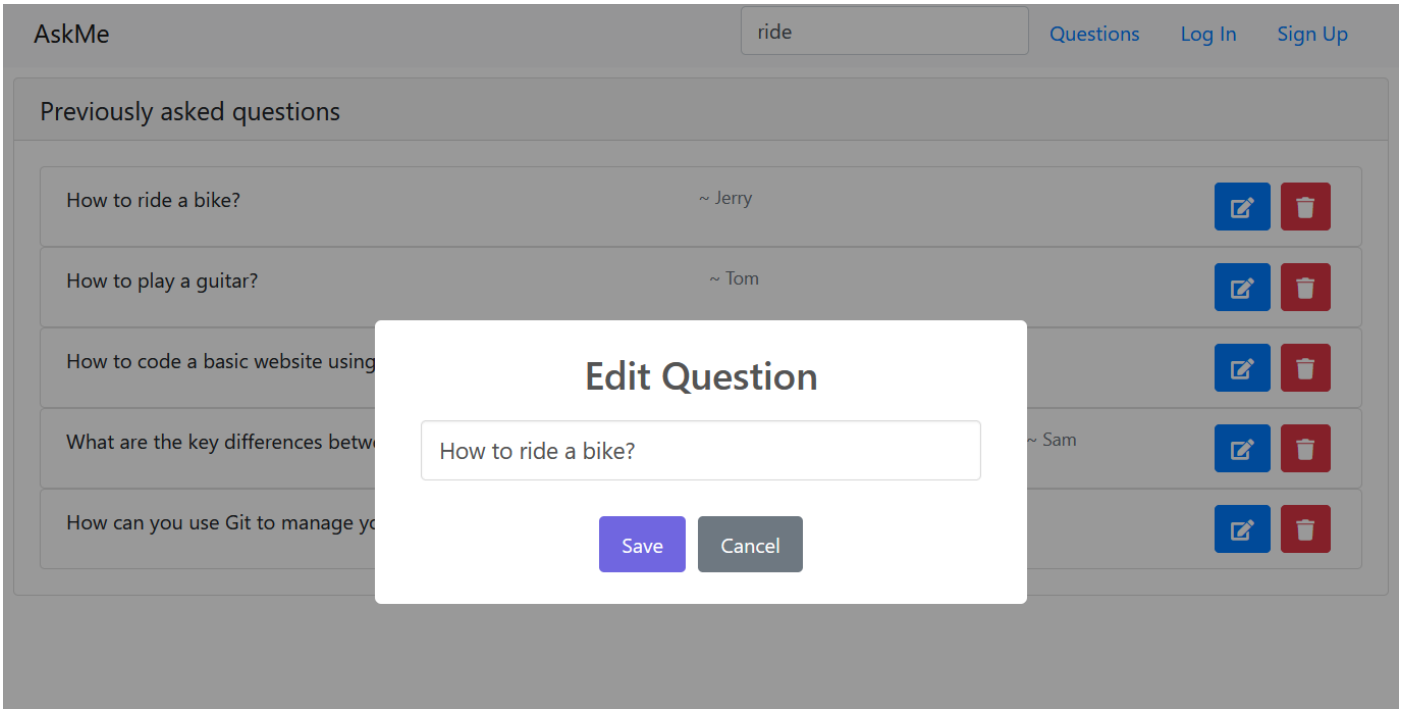
3.3 Post Question

Questions can be posted,

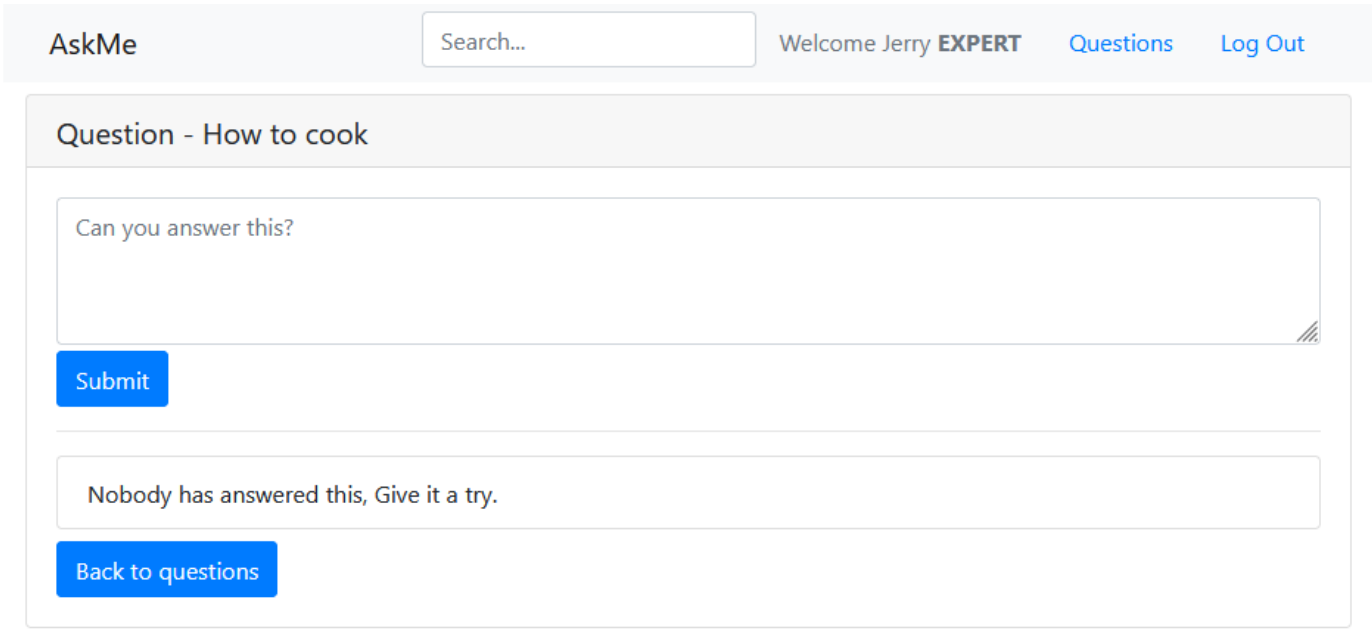


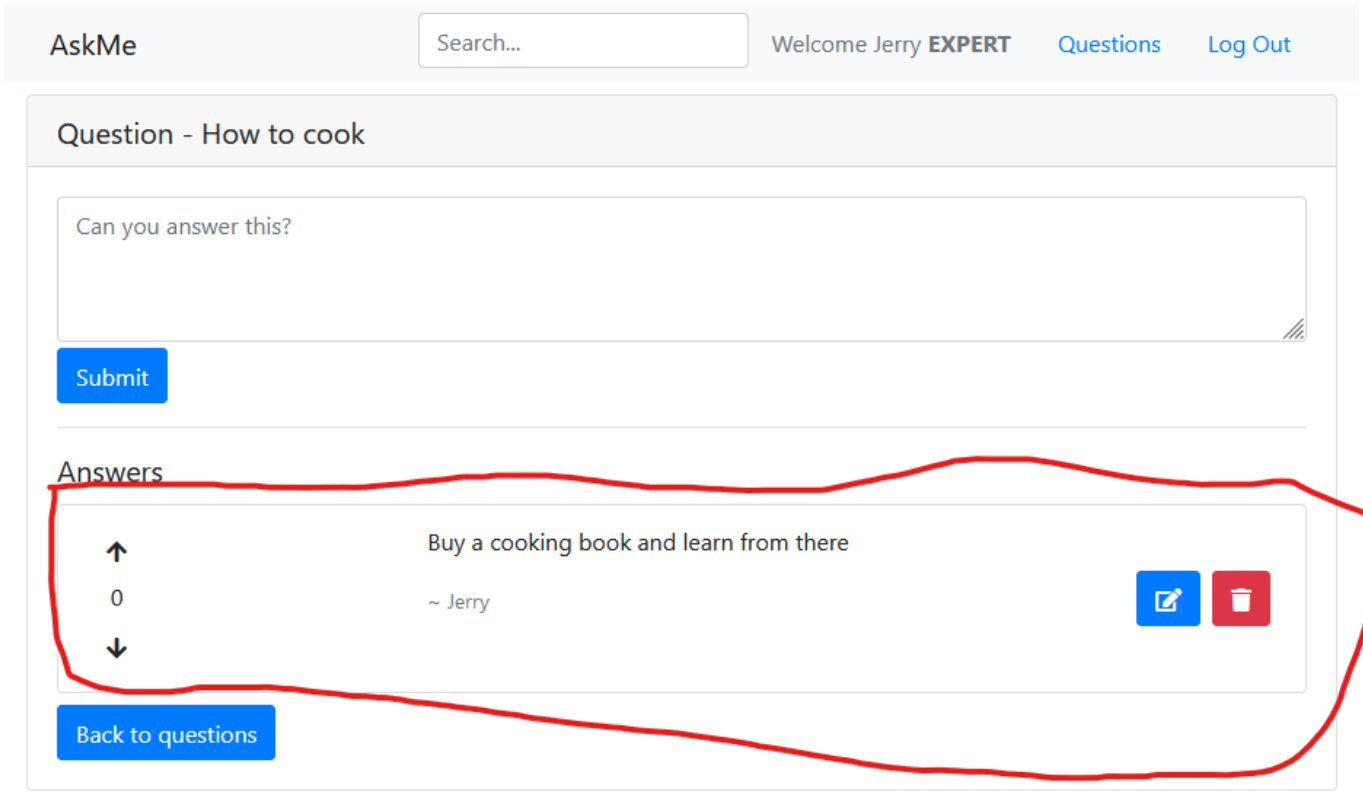


The questions can also be edited. SweetAlert2 to change the input box and to make the experience smoother.



3.4 Answer Question



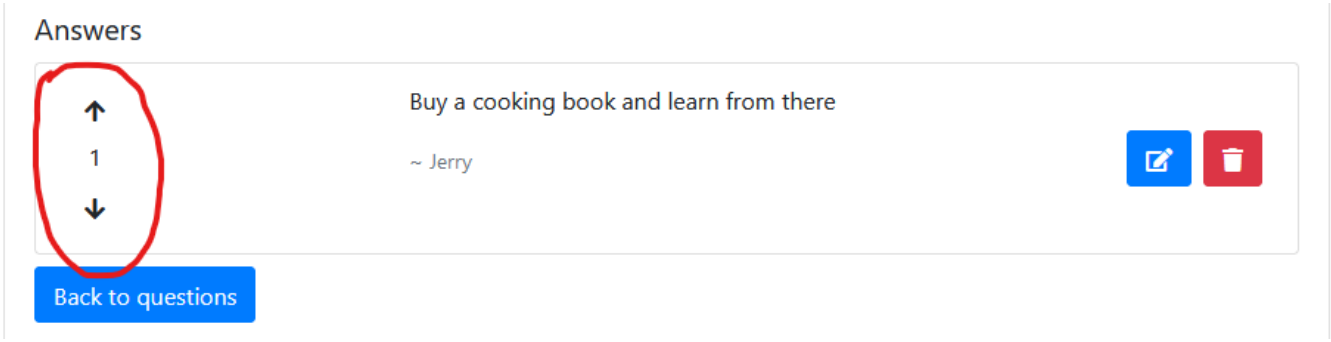


3.5 Filter out questions

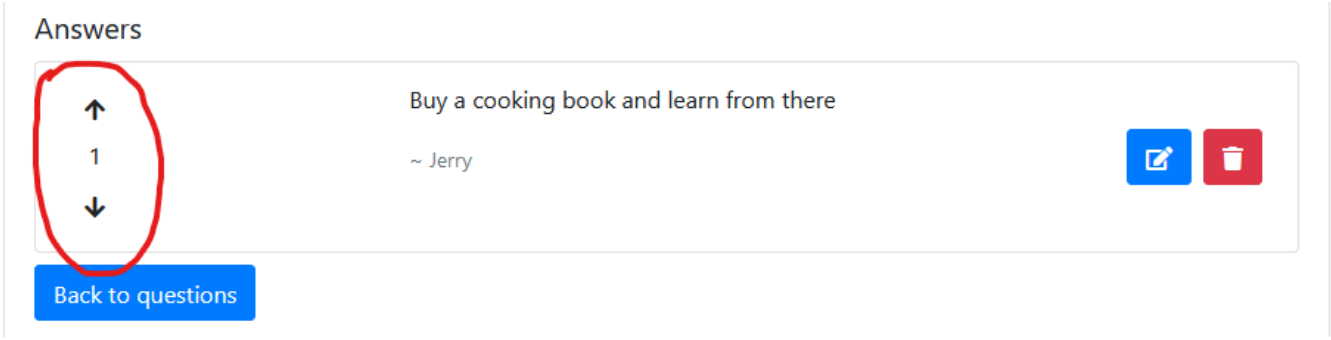
3.6 Approve answers

3.7 Rate answers

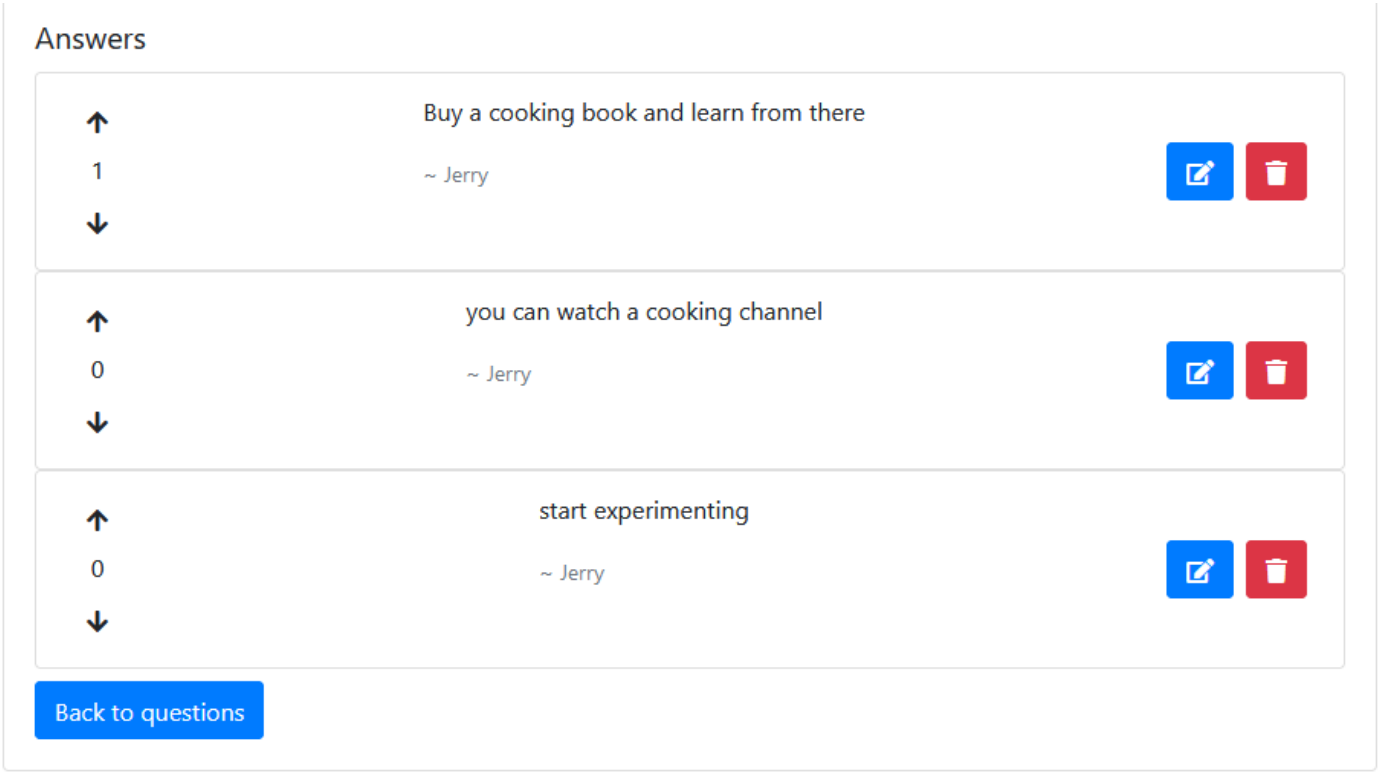
Answer can either be upvoted or downvoted.



3.8 Upvote



3.9 List answers



3.10 Search

3.11 Create comment

4 Design specifications

4.1 Inspired by

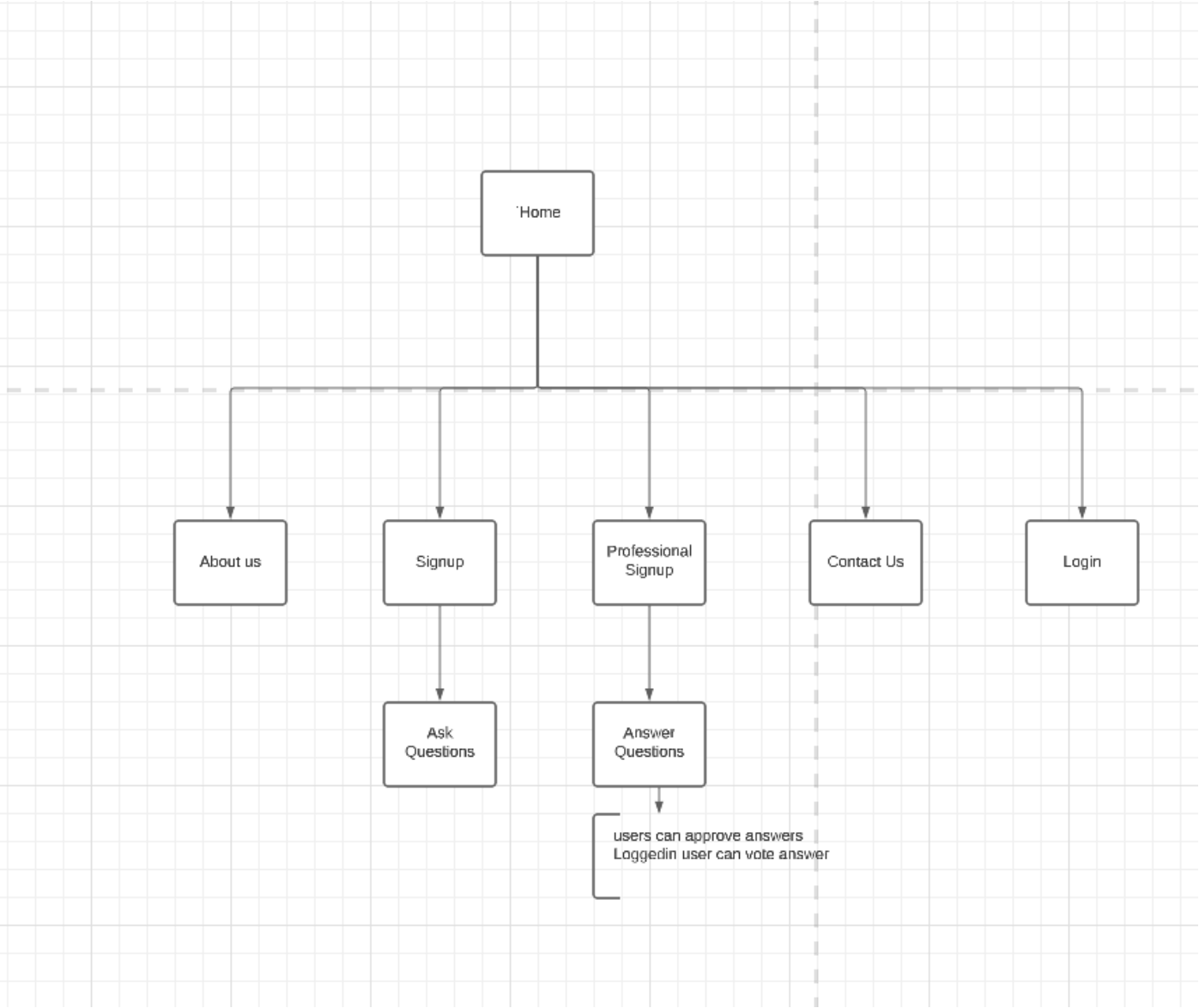
The website draws inspiration from the popular platform Stack Overflow, which serves as a model for its functionality and features. Like Stack Overflow, the website aims to create a community-driven platform where users can seek expert advice and assistance for problems.

Users can post questions on the platform, seeking help and solutions from experts and the community at large. They can provide detailed descriptions of their issues, including relevant information and attachments if needed. The website also allows for categorizing questions, enabling users to browse and filter topics of interest.

Different user roles exist within the platform to facilitate efficient interaction and administration. Expert users can not only post questions but also provide answers based on their knowledge and expertise. Admin users, in addition to asking and answering questions, have elevated privileges such as editing and deleting questions and answers. Normal users have limited capabilities, being able to comment and ask questions.

By embracing the Stack Overflow model, the website aims to create a user-friendly and collaborative environment enthusiasts can access valuable insights, engage in discussions, and find solutions to their problems. The platform encourages knowledge sharing and fosters a supportive community, much like the inspiration it draws from Stack Overflow.

4.2 Sitemap



4.3 Wireframe templates

4.4 Colour

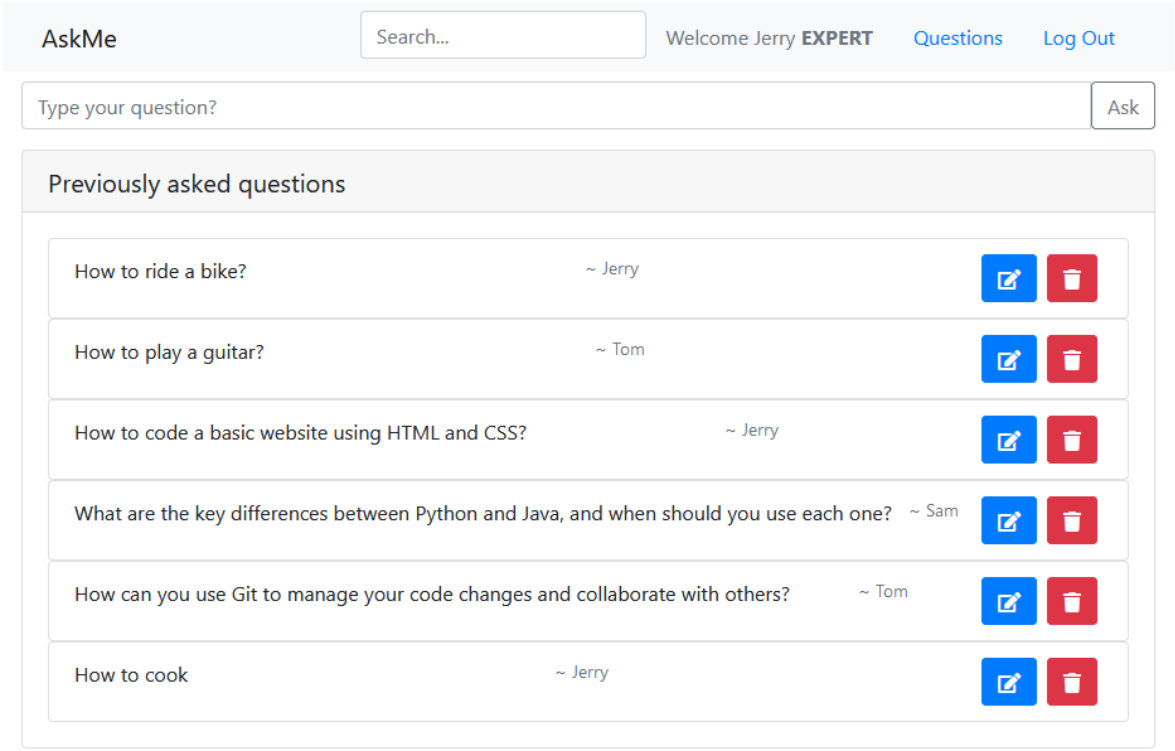
The web application features a clean and minimalist design, utilizing a color scheme primarily composed of white and grey tones. The use of these colours creates a simple and understated aesthetic, allowing the focus to be on the content and functionality of the application.

The predominance of white throughout the interface promotes a sense of clarity, openness, and neutrality. It enhances readability and ensures that the text and other elements stand out prominently against the background. The white color also conveys a feeling of cleanliness and professionalism, contributing to a pleasant user experience.

Complementing the white, subtle shades of grey are incorporated strategically to provide visual depth and contrast. Gray tones are often utilized for borders, dividers, or background elements, helping to delineate

different sections and create a sense of hierarchy within the interface. The restrained use of grey adds a touch of sophistication and elegance to the overall design.

By adopting a basic white and grey color scheme, the web application achieves a modern and timeless appearance. The simplicity and neutrality of these colours create a versatile foundation that can accommodate other design elements, such as icons, images, or accent colours, if necessary. The understated color palette ensures a visually pleasing and uncluttered interface that prioritizes usability and ease of navigation for users.



4.5 Fonts

The web application utilizes the widely accepted and versatile font, Open Sans, to ensure a consistent and professional typographic experience for users.

4.6 Photo or image treatment

4.7 General style

For the general style of the website, we have opted for a simple and classic black and white theme to keep the focus on the content and functionality of the site. The design is clean and straightforward, with minimal distractions and unnecessary embellishments. We have chosen a standard font and layout to ensure readability and ease of navigation for our users. This approach will make it easier for users to find and access the information they need quickly and efficiently. Overall, the general style of the website is sleek and professional, and we believe it will provide a great user experience.

5 Usage examples

The website could be used by small companies that would want to give customer service to their users while keeping moderation or even for personal use, you could end up on the website by looking up a similar question and find an answer to your question.

6 Application Structure

The application consists of several components, each with its own set of functionalities and responsibilities. The main components of the application are:

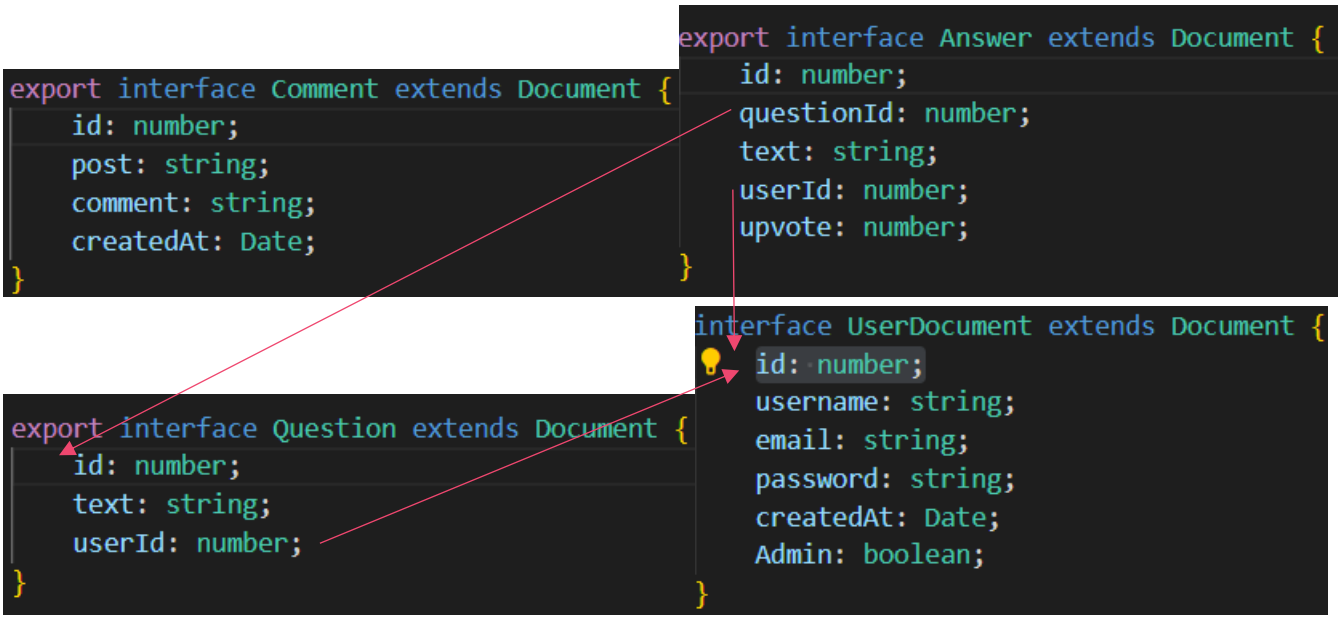
1. App Component: This is the root component of the application that serves as the entry point. It contains the navigation bar and the router-outlet component that loads the other components based on the current route.
2. Home Component: This component displays the landing page of the application and allows users to search for questions and answers.
3. Question Component: This component displays a single question and its associated answers. It allows users to upvote or downvote answers and submit their own answers.
4. Ask Question Component: This component provides a form for users to ask a new question.
5. Edit Question Component: This component allows users to edit an existing question.
6. Answer Component: This component displays a single answer and allows users to upvote or downvote the answer.
7. Ask Answer Component: This component provides a form for users to submit a new answer to a question.
8. Edit Answer Component: This component allows users to edit an existing answer.

In addition to the components, the application also includes several services that handle data retrieval and manipulation. These services include:

1. Question Service: This service is responsible for retrieving and manipulating questions from the backend API.
2. Answer Service: This service is responsible for retrieving and manipulating answers from the backend API.
3. User Service: This service handles user authentication and authorization, as well as user-related data.

The application structure is designed to be modular and scalable, making it easy to add new components and services as the project evolves.

7 Database Structure



8 The project timeline

- Reading documentation
- Doing the Tour of Heroes tutorial
- Setting up the SiteMap
- Setting up the tasks, ordering them and attributing them
- Adding new small tasks as time went on and new problems/ideas came along
- Finishing up the project and putting it up online

9 Reflection on how the project went

Overall, the project was a great learning experience, especially because it was a group project. Learning Angular was also very interesting as it uses the fundamentals we've worked on during our studies while implementing its own concepts on top of it. The project started slowly as we were slowly learning this new technology though mock projects and documentation, but it was a dynamic experience where we would complete each other's flaws

GITHUB LINK

https://github.com/ApexPlayground/HomeOwnersAngularApp