DISC DETECTIVE

Capstone project by Adam Sagar

Introduction

Purpose

- *What is the problem or the opportunity that the project is investigating?*

This project aims to establish a platform for disc golfers to connect and retrieve their lost discs.

Disc golf is a sport similar to traditional golf, but instead of using balls and clubs, players throw frisbee-like discs into baskets.

This is how to disc golf

This is how not to disc golf

Lost discs are a common occurrence in disc golf, and unfortunately, they often go unretrieved. This is largely due to the absence of any contact information on the disc. Even when such information is provided, it may become outdated or illegible over time, further reducing the chances of a successful return.

- *Why is this problem valuable to address?*

It can disrupt the flow of the game, resulting in lost time and frustration for the players.

- *What is the current state (e.g. unsatisfied users, lost revenue)?*

Losing discs can be expensive for players, and it can also be upsetting if the disc holds sentimental value.

- *What is the desired state?*

More discs being returned to their owners. This is possible when more people are involved in the search. The likelihood of a successful return increases, as someone else may come across the disc by chance and facilitate its return.

- *Has this problem been addressed by other projects? What were the outcomes?*

The website lostandfound.dzdiscs.com/ helps people locate lost discs, but unfortunately, its services are only located in the USA. I intend to launch a similar platform in Christchurch, New Zealand, with eventual expansion nationwide. This new platform will provide more extensive filtering options for locating lost discs. Users will also have the option to create posts stating they have lost a disc, allowing for a more proactive role in their search.

Industry / domain

- *What is the industry/ domain?*

Sports and recreation industry

- *What is the current state of this industry? (e.g. challenges from startups)*

Experiencing significant growth in the last few years. Many people took it up as a safe way to socialise with their friends during the pandemic. There are now 50 courses and 20,000 active players in New Zealand.

The entry barrier is low because there is minimal equipment required and many of the courses are free to play.

More and more courses are being built in existing parks and recreational areas. This has resulted in a higher frequency of lost discs. As a result, there is an increased need for a platform like this.

- *What is the overall industry value-chain?*

From a social perspective the value of disc golf is primarily in the enjoyment that it provides.

- *What are the key concepts in the industry?*

This project will focus on the social side of disc golf. Providing a platform that enhances people's experience can only be beneficial for the growing sport.

- *Is the project relevant to other industries?*

The concept could apply to other contexts where there is a community driven approach to lost and found item recovery. These could include personal items or pets.

Stakeholders

- *Who are the stakeholders? (be as specific as possible as to who would have access to the software)*

Disc golfers looking for their lost disc.

Disc golfers looking to return a found disc.

Members of the public who come across a lost disc.

Course operators (if privately owned).

- *Why do they care about this software?*

Disk golfers - specialised discs aren't cheap and it's not uncommon to lose multiple a round. Some of these discs may also hold sentimental value to the player.

Members of the public - the desire to do the right thing or as a way to "pay it forward".

Course operators - Christchurch has two privately owned courses. Promoting or integrating the platform into their site could lead to a better experience for their customers.

- *What are the stakeholders' expectations?*

To be able to find or return a lost disc.

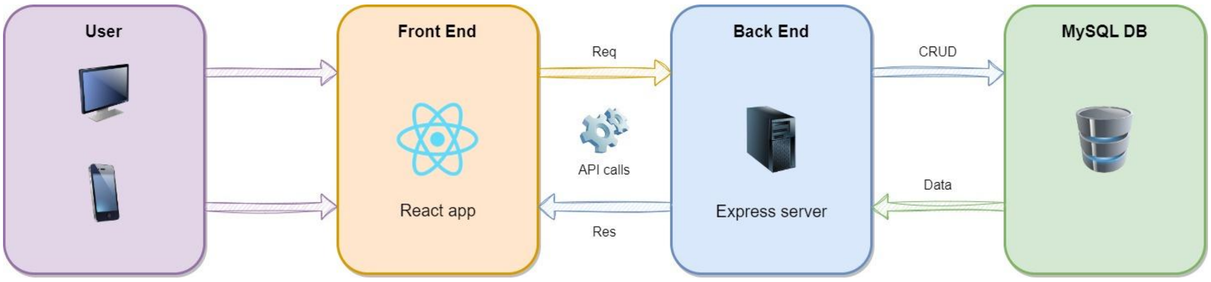
An intuitive UI that enables them to easily search for or create a post about a lost disc.

Contact the person who lost or found a specific disc.

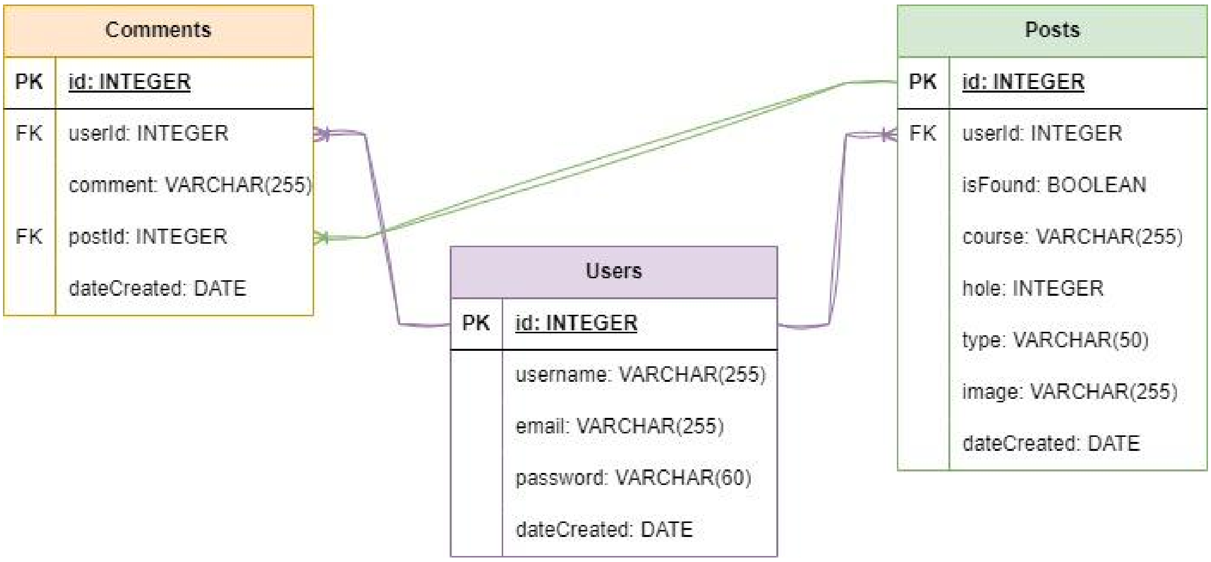
Product Description

Architecture Diagram

Include a diagram of the building blocks of the design including users and how they interact with the product.



Database Diagram



Comments	
PK	id: INTEGER
FK	userId: INTEGER
	comment: VARCHAR(255)
FK	postId: INTEGER
	dateCreated: DATE

Users	
PK	id: INTEGER
	username: VARCHAR(255)
	email: VARCHAR(255)
	password: VARCHAR(60)
	dateCreated: DATE

Posts	
PK	id: INTEGER
FK	userId: INTEGER
	isFound: BOOLEAN
	course: VARCHAR(255)
	hole: INTEGER
	type: VARCHAR(50)
	image: VARCHAR(255)
	dateCreated: DATE

User Stories

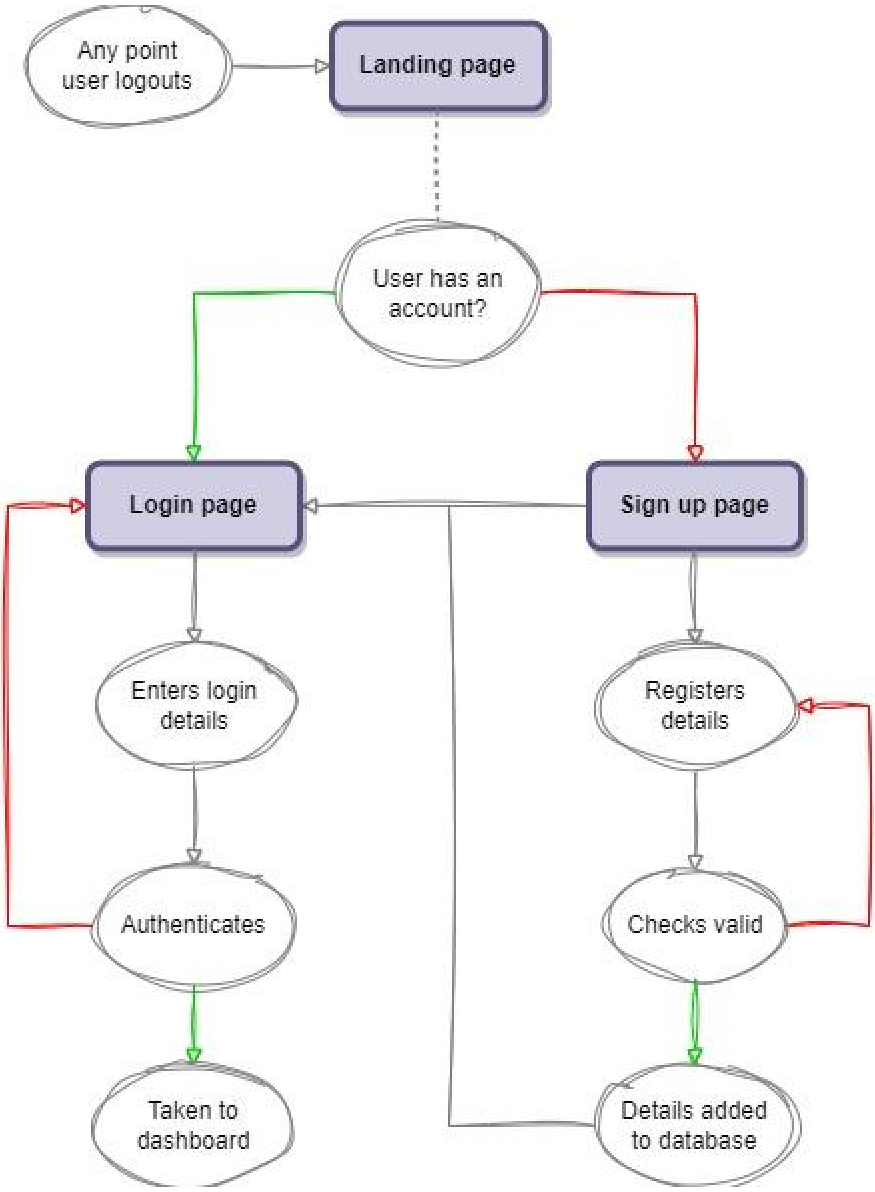
#	User Story Title	User Story Description	Priority	Additional Notes
1	Sign up	As a user, I want to be able to sign up to access the site.	High	Achieved
2	Sign in	As a user, I want to be able to sign in to create/interact with posts.	High	Achieved
3	Authentication	As a user, I want my information to be secure. Achieved with: ● Token ● Encrypted passwords	Low	Token became out of scope Achieved password encryption
4	View posts	As a user, I want to be able to view a list of found discs for two reasons: ● To see if someone has found my disc. ● To see if someone is looking for the disc I've found.	High	Achieved
5	Dashboard	As a user, I want to have a dashboard where I can easily view and manage my posts.	High	Achieved
6	Filter posts	As a user, I want to be able to filter the posts, first by lost or found and then course, hole and type.	High	Achieved
7	Comment on posts	As a user, I need the ability to comment on a post, so I can organise retrieving my disc.	High	Achieved
8	Create posts	As a user, I want to create posts for two reasons: ● For a disc I've found ● Taking a proactive approach in finding my lost disc.	High	Achieved
9	Delete posts	As a user, I want to be able to delete my post once the disc has been recovered.	High	Achieved

10	Subscribe to posts	As a user, I want to be able to subscribe to posts I've commented on so the thread is easy to access.	Med	Became out of scope but will implement at a later date.
11	Mobile-friendly	As a user, I need the site to be mobile-friendly because I may access it while on the course.	Med	Achieved
12	Create play-with posts	As a user, I want to be able to find new people to play with.	Low	While not essential, adding social media features is a potential future direction for the site.

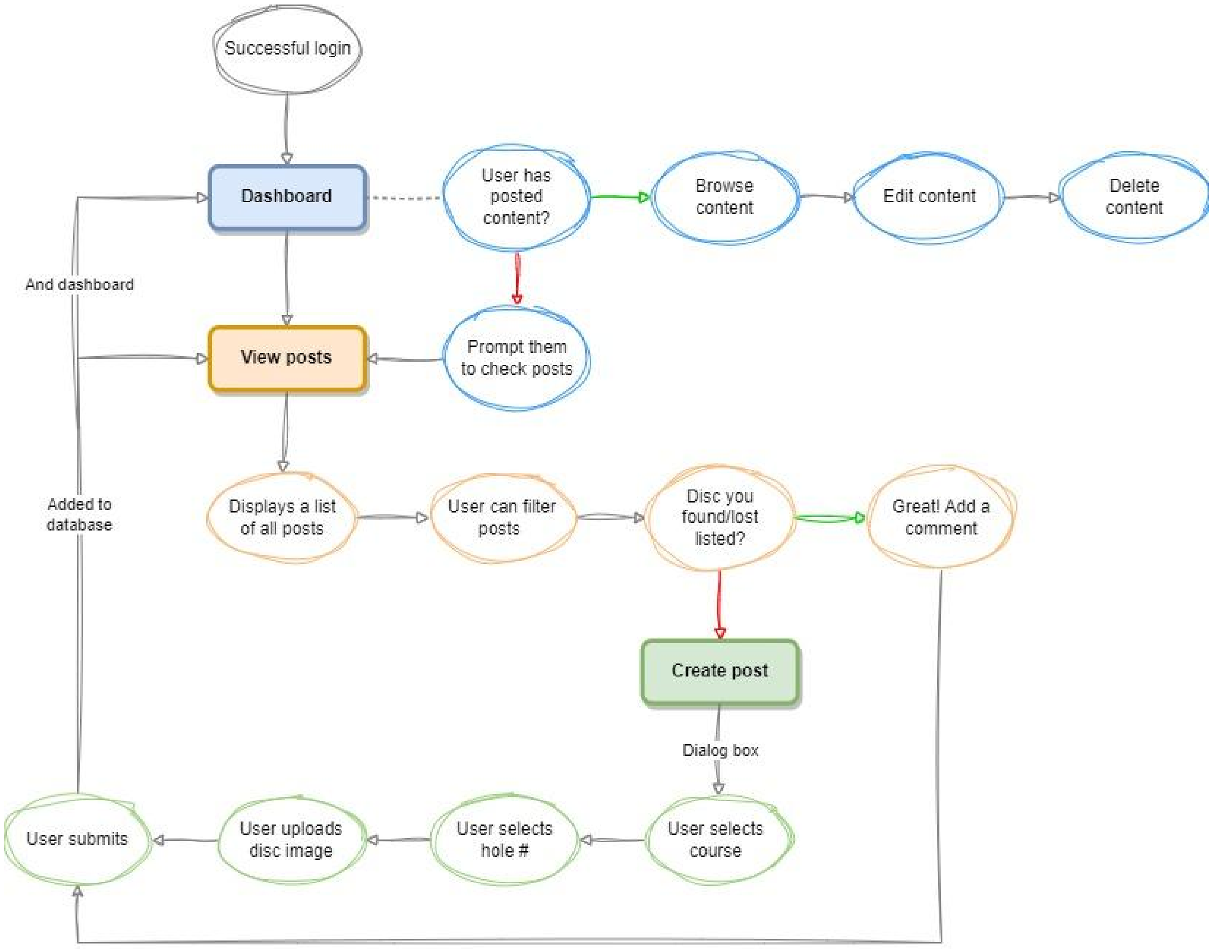
User Flow

Present as a flow diagram the steps a user may make in interacting with the software.

Login and Sign up



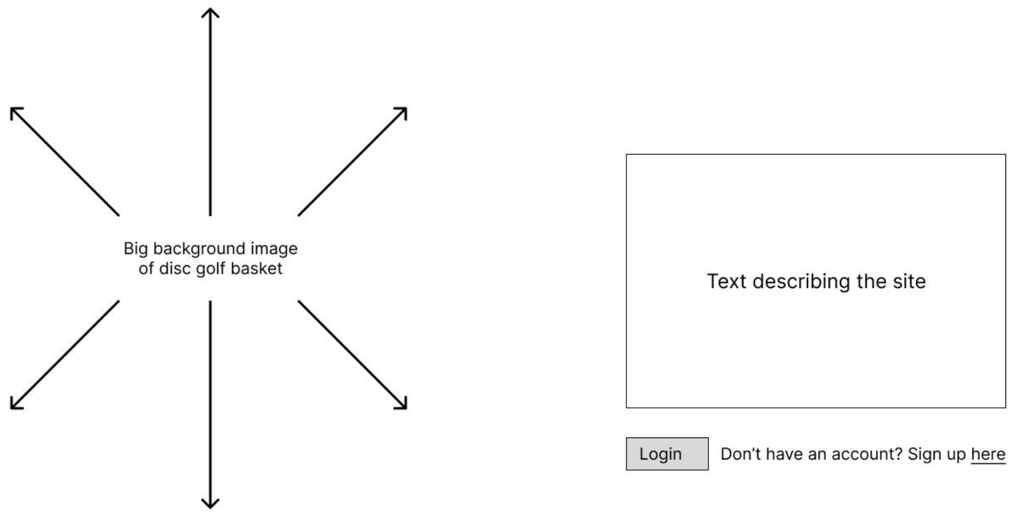
Main Site Flow

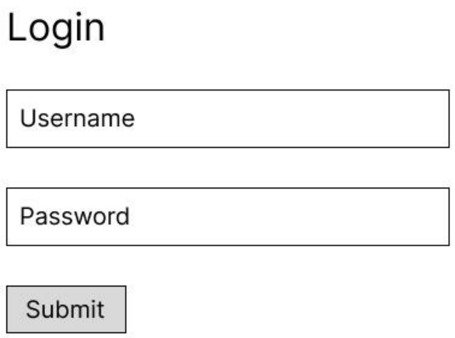


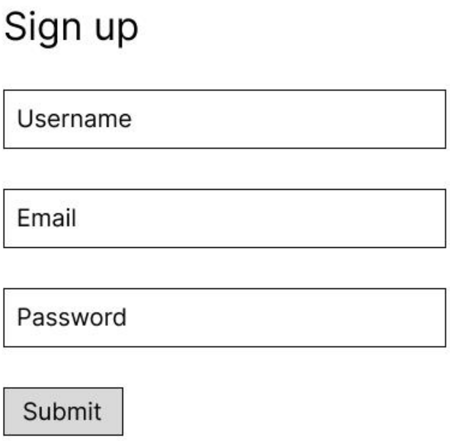
Wireframe Design

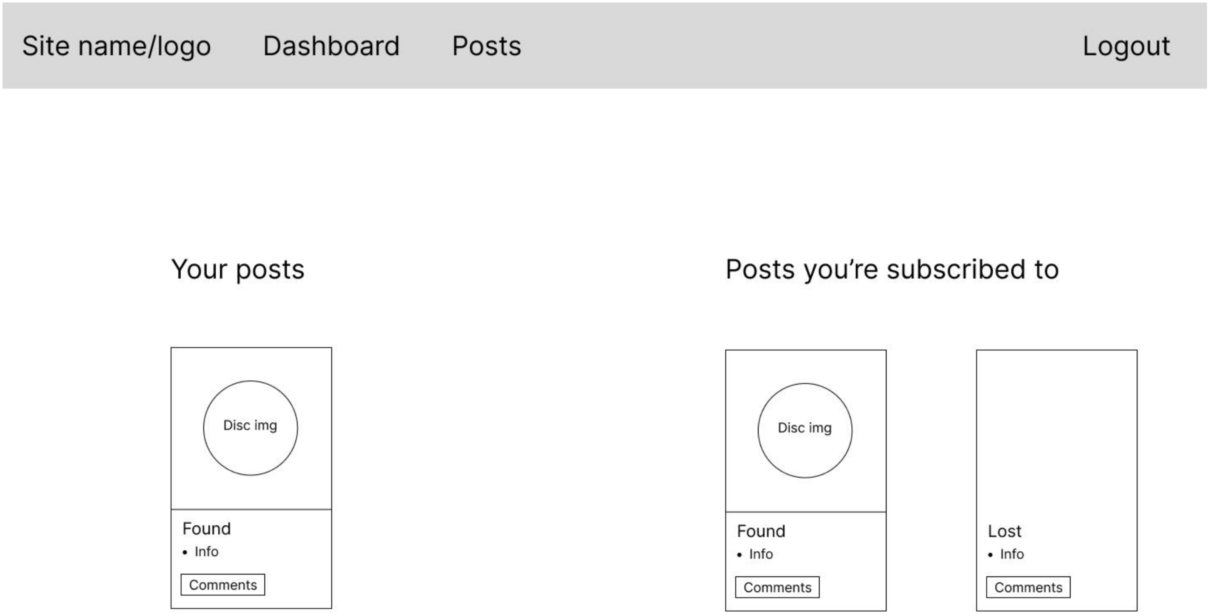
Show elements of the user interface, either manually or via a tool such as Figma.

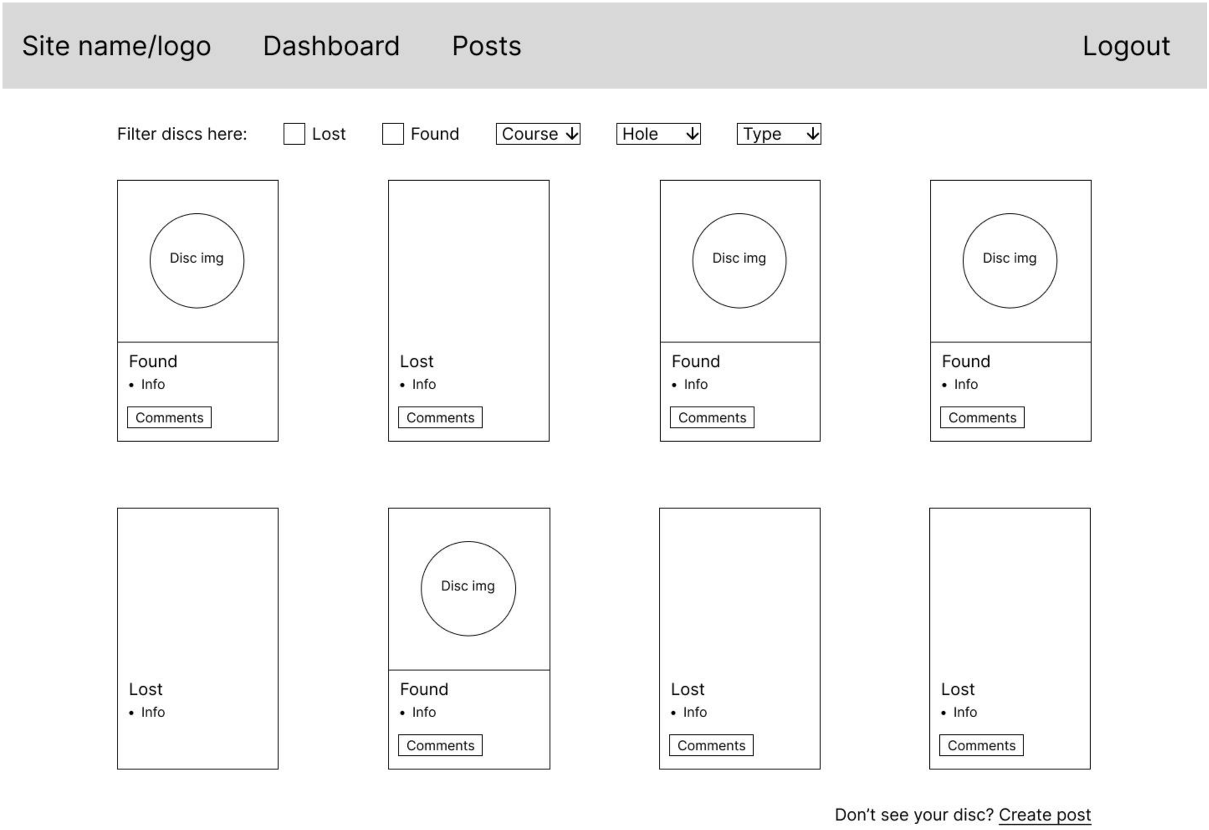
Lo-fi Wireframes



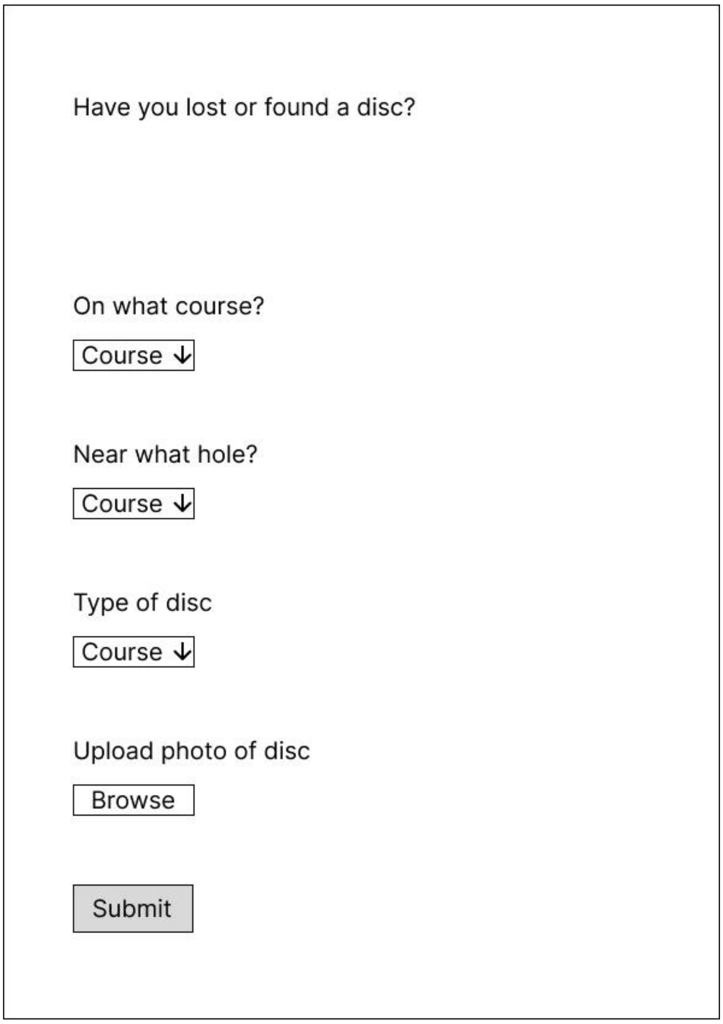








Don't see your disc? Create post



Have you lost or found a disc?

On what course?

Course ↓

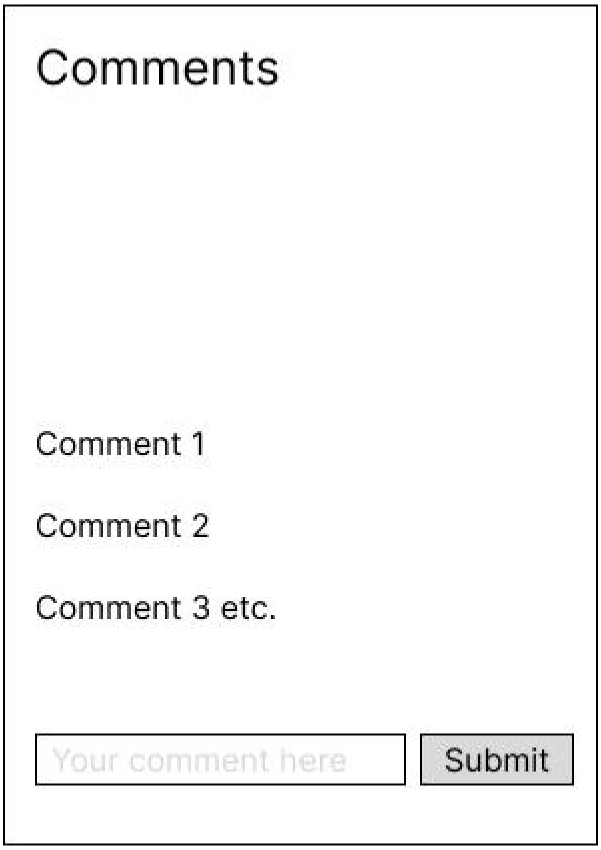
Near what hole?

Course ↓

Type of disc

Course ↓

Upload photo of disc



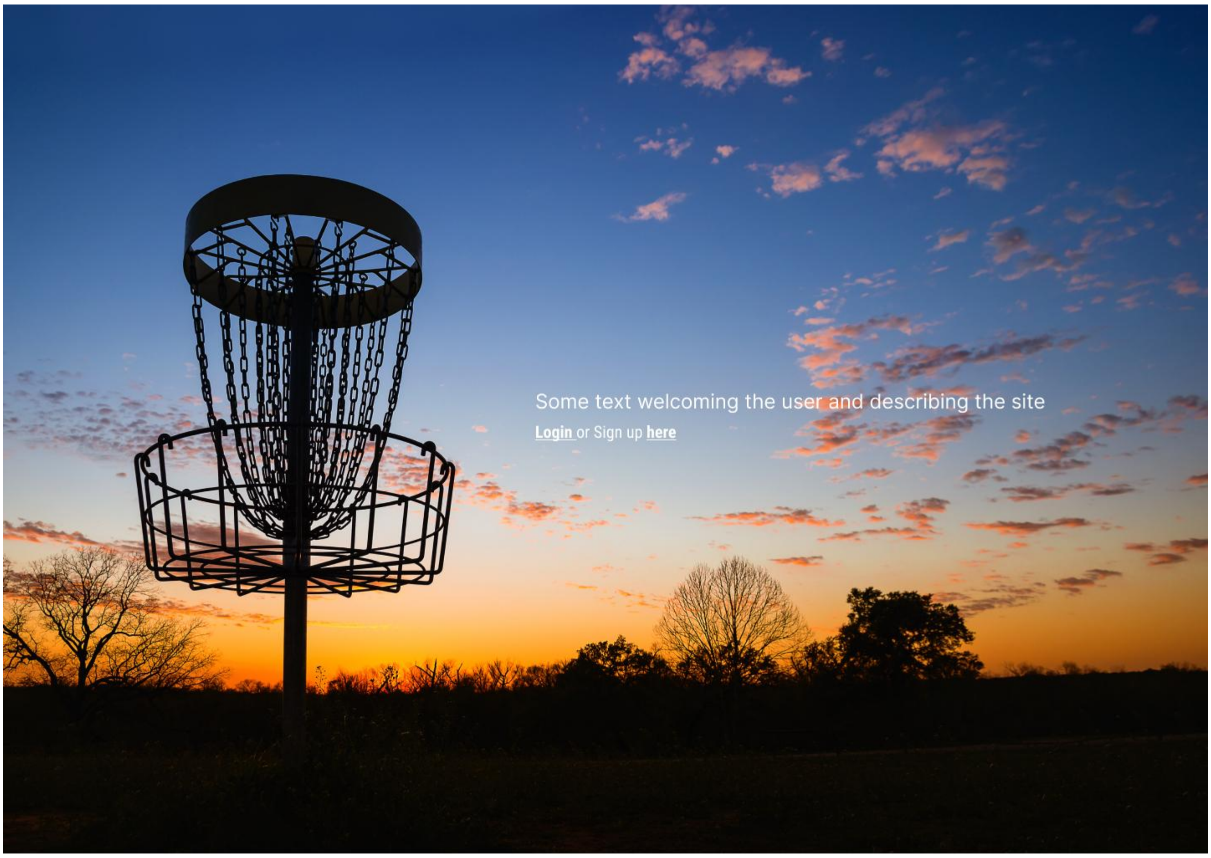
Comments

Comment 1

Comment 2

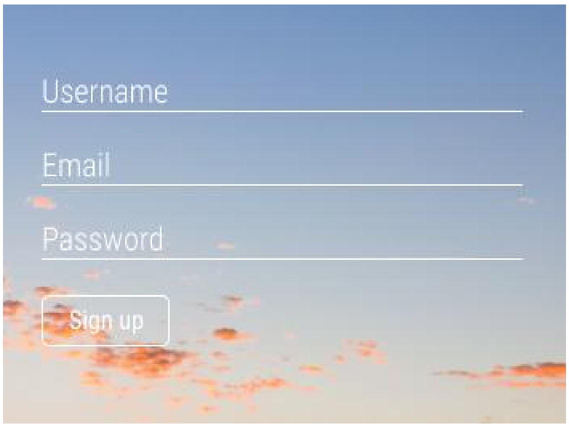
Comment 3 etc.

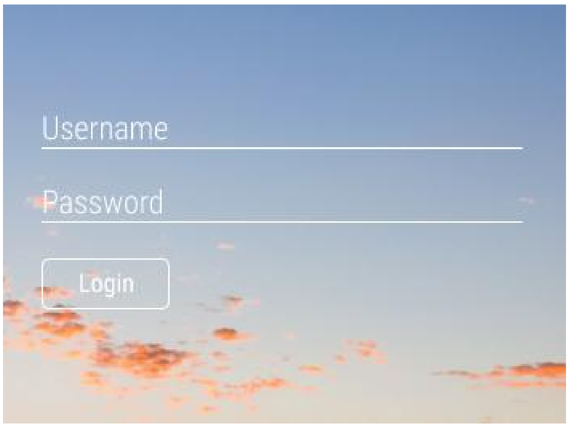
Hi-fi Wireframes

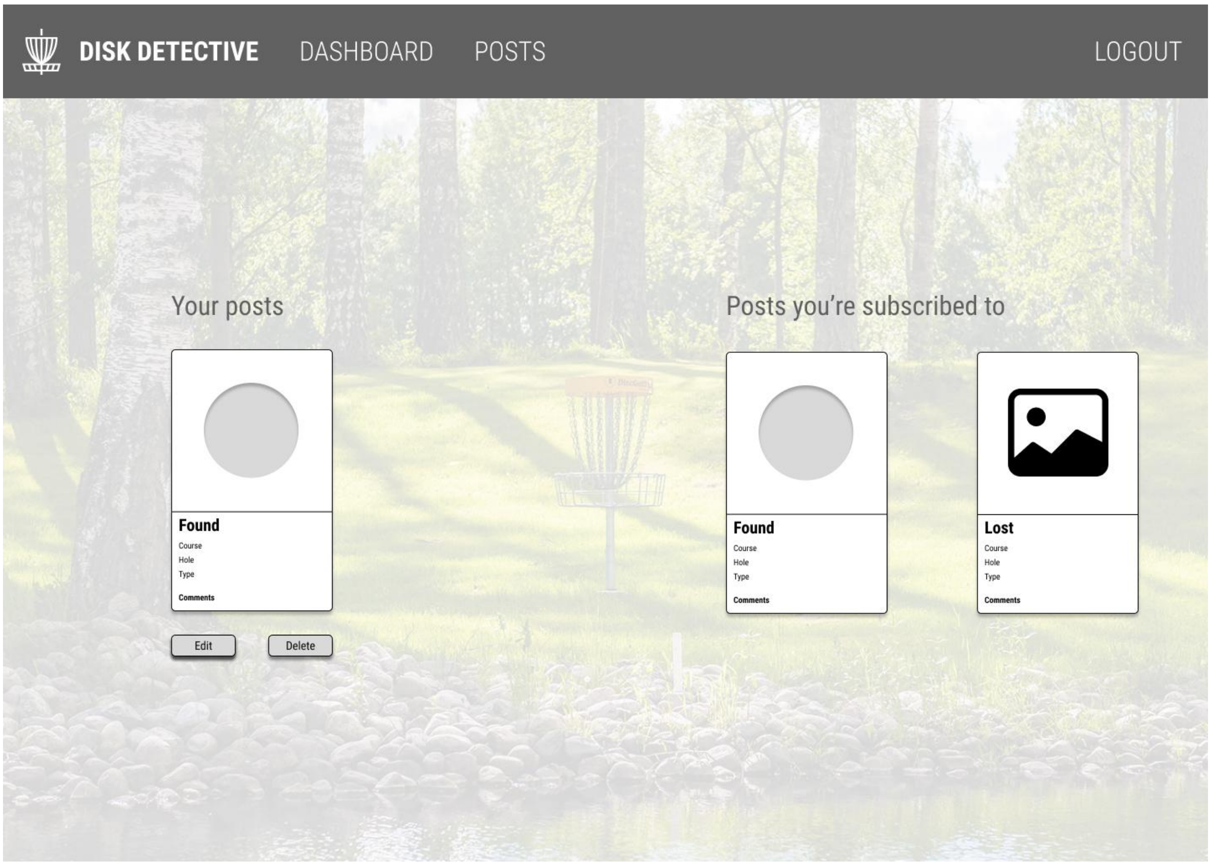


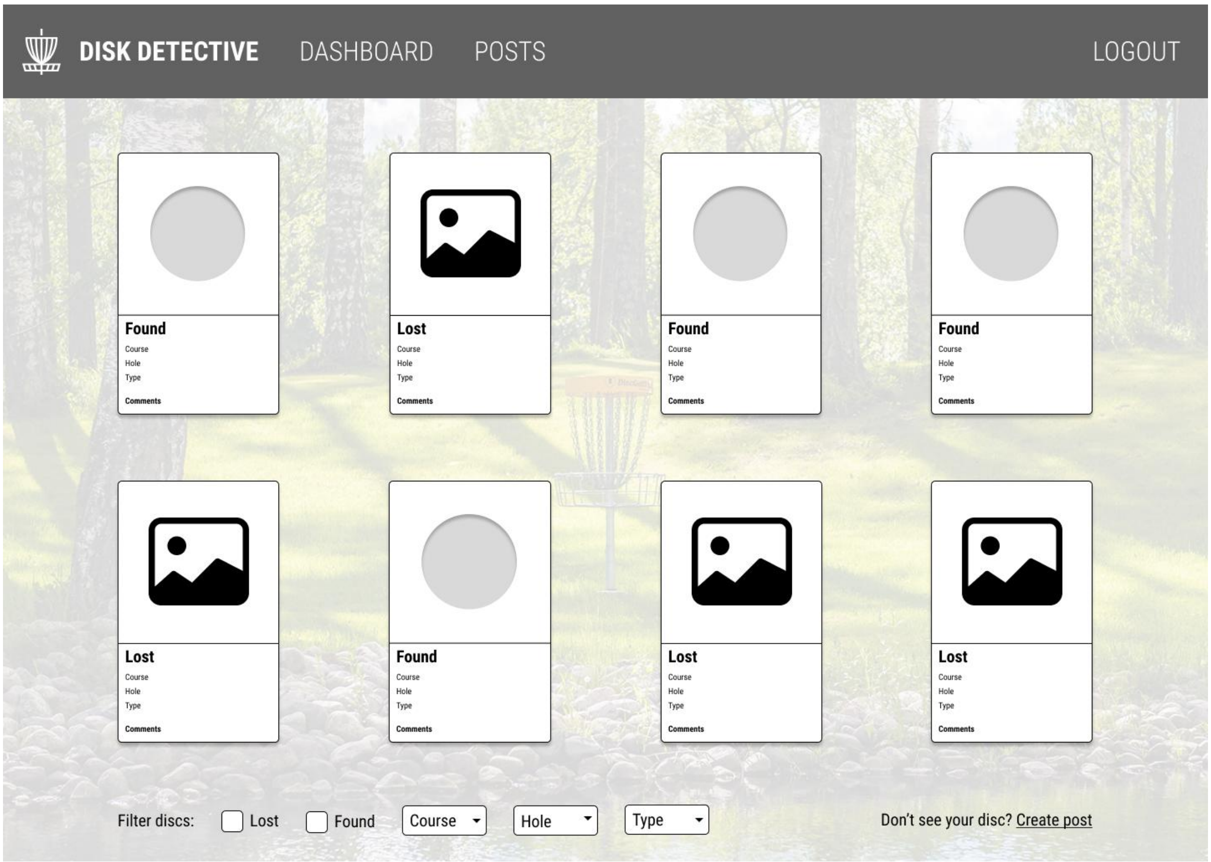
Some text welcoming the user and describing the site

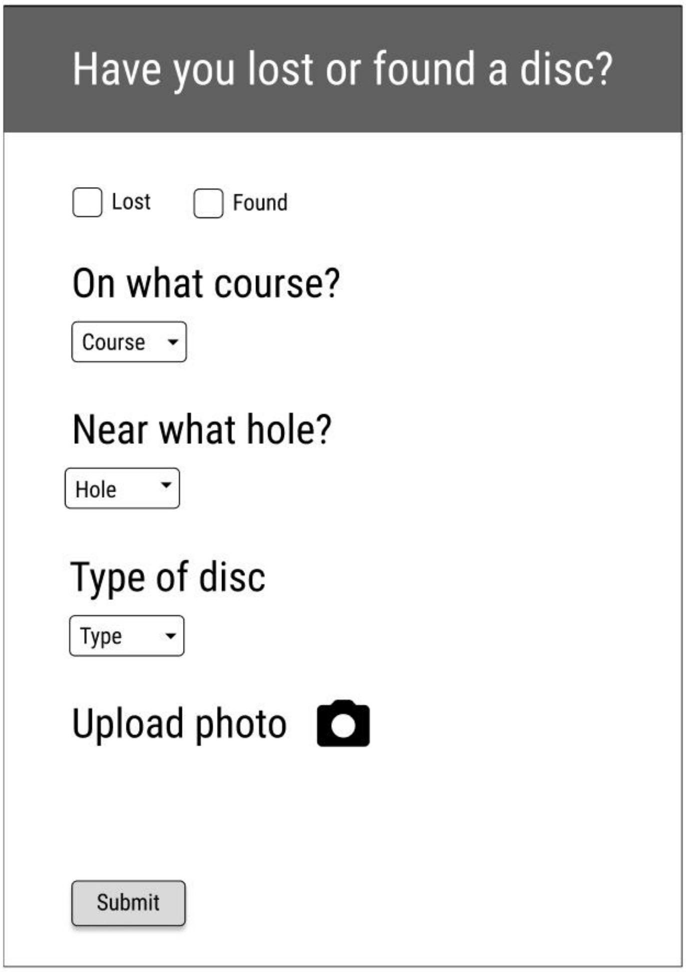
Login or Sign up **here**






DISK DETECTIVE
DASHBOARD
POSTS
LOGOUT
Your posts
Posts you're subscribed to
Found
Course
Hole
Type
Comments
Edit
Delete
Found
Course
Hole
Type
Comments
Lost
Course
Hole
Type
Comments


DISK DETECTIVE
DASHBOARD
POSTS
LOGOUT
Found
Course
Hole
Type
Comments
Lost
Course
Hole
Type
Comments
Found
Course
Hole
Type
Comments
Found
Course
Hole
Type
Comments
Lost
Course
Hole
Type
Comments
Found
Course
Hole
Type
Comments
Lost
Course
Hole
Type
Comments
Lost
Course
Hole
Type
Comments
Filter discs:
□ Lost
□ Found
Course
Hole
Type
Don't see your disc? Create post



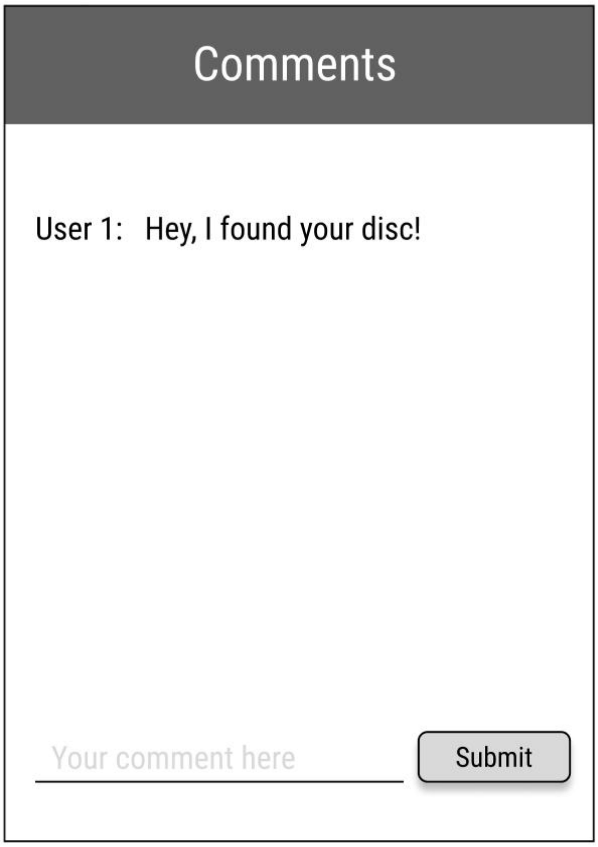
Have you lost or found a disc?

On what course?

Near what hole?

Type of disc

Upload photo



Comments

User 1: Hey, I found your disc!

Open Questions / Out of Scope

- *What features are considered out of scope?*

Notifications - since the main purpose of this site is disc retrieval, email notifications for post comments would be helpful because most users would not check the site daily.

Geolocation - ambitious but allowing users to create a post at their current location on the course would be convenient, as they wouldn't have to remember to do it later. A more viable option may be providing a map of each course where users can place a marker themselves.

Social media - potential to expand into the realm of social media as a platform for disc golf enthusiasts to interact and engage with each other.

Non-functional Requirements

- *What are the key security requirements? (e.g. login, storage of personal details, inactivity timeout, data encryption)*

The database encrypts passwords for secure storage and transfer.

Once private messaging is added, it would be important to consider a session timeout to prevent any personal information shared between users, such as an address for retrieval, from being exposed for an extended period of time.

- *How many transactions should be enabled at peak time?*

The most popular park listed can have up to 500 people going through it in a single day. While it's unlikely they would all be online at the same time the site would ideally support this number. In the future this number will increase as more cities/courses are added so having scalable infrastructure is an important consideration.

- *How easy to use does the software need to be?*

It should have a familiar flow to similar sites so it feels intuitive to the user.

- *How quickly should the application respond to user requests?*

Like most sites, ideally as quick as possible to ensure a smooth experience.

- *How reliable must the application be? (e.g. mean time between failures)*

Ideally it would be up 100% of the time except for maintenance.

- *Does the software conform to any technical standards to ease maintainability?*

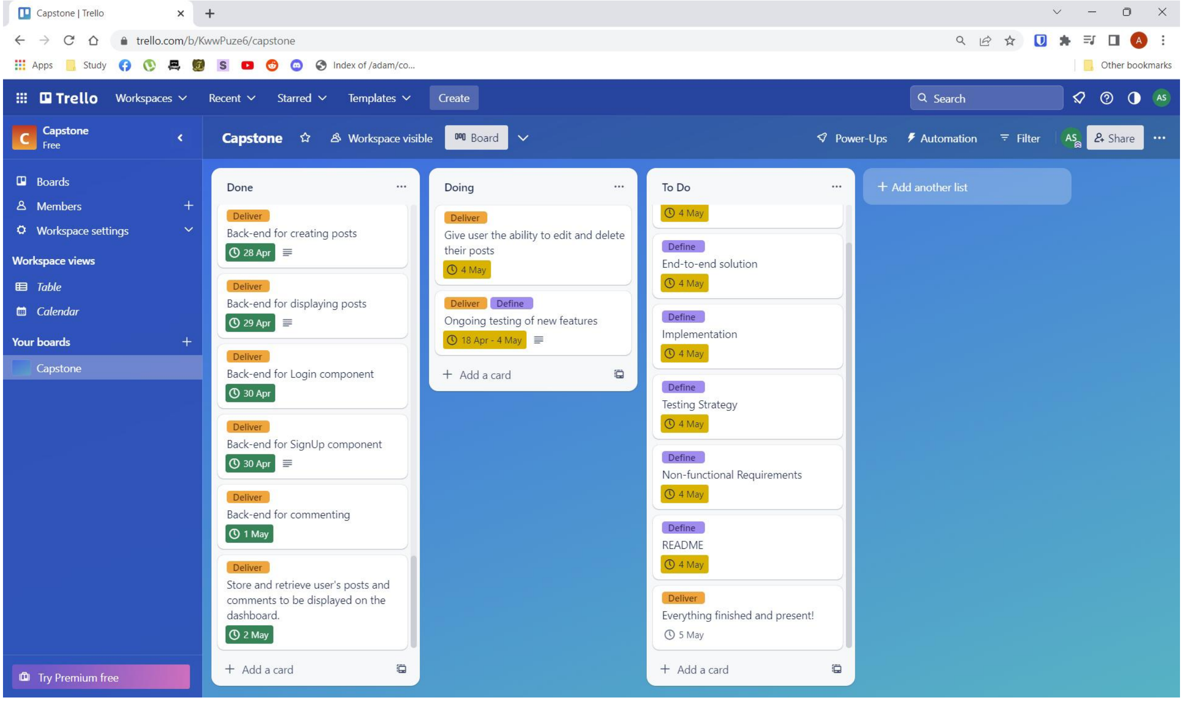
The site is designed with a separation of the front and back-end, and employs the MVC structure to ensure easy maintenance and the ability to incorporate new features in the future.

Project Planning

Include a Gantt chart or screenshot of a Trello board showing key milestones (with dates) to complete the project.

In order to meet the project deadline, I used Trello to define and categorise the key features into manageable tasks with corresponding completion dates. While this approach worked well overall, it did reveal one issue: working on the front and back-end at the same time. Some features depended on the back-end of other features being complete in order to test them, without that it hindered the testing process. This will be taken into consideration for all future projects.

https://trello.com/b/KwwPuze6/capstone



Done

- Deliver — Back-end for creating posts — 28 Apr
- Deliver — Back-end for displaying posts — 29 Apr
- Deliver — Back-end for Login component — 30 Apr
- Deliver — Back-end for SignUp component — 30 Apr
- Deliver — Back-end for commenting — 1 May
- Deliver — Store and retrieve user's posts and comments to be displayed on the dashboard. — 2 May

Doing

- Deliver — Give user the ability to edit and delete their posts — 4 May
- Deliver, Define — Ongoing testing of new features — 18 Apr - 4 May

To Do

- 4 May
- Define — End-to-end solution — 4 May
- Define — Implementation — 4 May
- Define — Testing Strategy — 4 May
- Define — Non-functional Requirements — 4 May
- Define — README — 4 May
- Deliver — Everything finished and present! — 5 May

Testing Strategy

- *What were steps undertaken to achieve product quality?*

Every feature was tested as it was created and retested whenever a new feature was added to ensure that no further bugs were introduced.

All of the API endpoints were established early in the development process and tested thoroughly with Postman to ensure they were able to receive the data correctly. This helped to attribute any future errors to either capturing the data in the front-end or the processing of it in the back-end.

Error handling was used where appropriate to prevent the entire site from crashing and to log errors for reviewing.

- *How was each feature of the application tested?*

Signup and login

Console logging was used to verify that the state variables responsible for capturing the user input were functioning correctly prior to being sent to the back-end.

Try-catch blocks were used in the back-end to validate a users' login credentials against the database, producing distinct error messages depending on whether the username existed, the password was correct, or all the mandatory fields were filled out.

Creating Posts

Images weren't displayed on posts after being uploaded to the server via a user-filled form. To fix this issue it was necessary to switch from using an object to send the data to using FormData instead. This was because images can only be sent as binary data in the body of the request, which can only be achieved by using FormData.

Editing Posts

Submitting an edit to a post wasn't triggering a re-render of the component. This was due to the use of FormData, which converts everything to a string when sending input to the server. As some of the fields contained Booleans and numbers, there were conflicts with types, preventing the re-render. The issue was fixed by converting it back to its original type.

- How did you handle edge cases?

Select dropdowns and radio buttons were used where possible to limit the available options to users. This helps to ensure that the data in the database is consistent and within a specific range.
The signup component uses a pattern prop to restrict the characters a user can input into the text fields, preventing the possibility of data that could corrupt the database. The email input is set to “email” type to ensure that users input it in the correct format. All inputs are also set to “required” to guarantee that a complete form is submitted to the database to maintain consistency.

Implementation

- *What were the considerations for deploying the software?*

The site will be hosted within a Docker container and deployed on AWS in the future, but for now it is run locally.

Setup Instructions

Pre-requisites:

- Visual Studio Code
  https://code.visualstudio.com/download

- Node.js
  https://nodejs.org/en/download

- MySQL/Workbench
  https://www.mysql.com/downloads/
  https://www.mysql.com/products/workbench/

Application Setup:

1. Clone the repo at https://github.com/adam-sagar/CAPSTONE in VS Code

2. Open up a terminal and type

   ```
   cd CAPSTONE
   cd front-end
   npm install
   ```

   This will install all the dependencies for the front-end

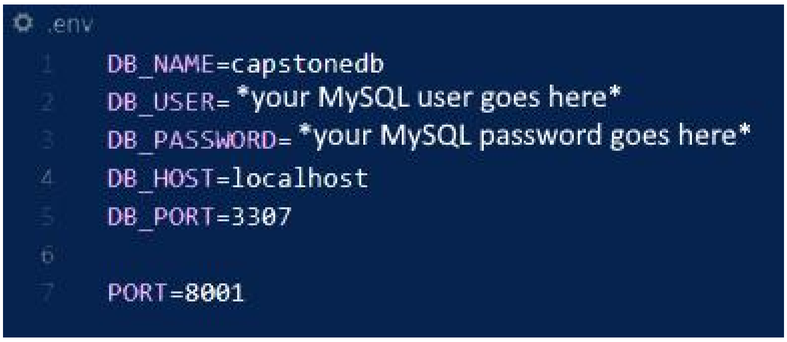
3. Open up another terminal and type

   ```
   cd CAPSTONE
   cd back-end
   npm install
   ```

   This will install all the dependencies for the back-end

Database Setup:

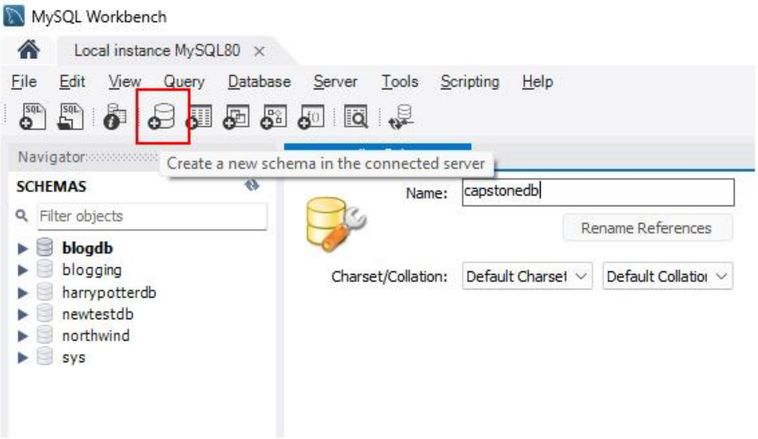
1. Create a .env file in CAPSTONE/back-end and add the following:



```
DB_NAME=capstonedb
DB_USER= *your MySQL user goes here*
DB_PASSWORD= *your MySQL password goes here*
DB_HOST=localhost
DB_PORT=3307

PORT=8001
```

2. Open MySQL Workbench and create a new schema called capstonedb. You can change the name but make sure to update your .env file. The database will be populated as you create an account, make posts and add comments.



Create a new schema in the connected server

Name: capstonedb

Charset/Collation: Default Charset Default Collation

Running the app:

1. Go back to your terminals in VS Code

2. Type *npm start* in CAPSTONE/back-end

3. Type *npm run dev* in CAPSTONE/front-end

4. Hold ctrl and click ➜ Local: http://localhost:5173/ to launch the app

End-to-end solution

- *How well did the software meet its objectives?*

The project has successfully met all of its main objectives. Users can sign up and easily search through the database of lost and found discs. If the disc they are looking for or trying to return is not already listed, they can quickly create a new post for it. Other users can then comment on the post, starting the process of locating and retrieving the disc.

Something that fell out of scope but is still important is adding any posts a user has commented on to their dashboard. This would provide easy access to any future replies or updates regarding those posts.

Another important issue to address is maintaining users' privacy when communicating with one another. Currently, every user can view the entire comment thread for a post. A user's security could be compromised if they were to share personal details like an address in a public thread. Adding a private messaging feature would prevent these risks and make for a more secure platform.

References

- *Where is the code used in the project? (link to GitHub)*

Code and documentation can be found at the following GitHub repository:
https://github.com/adam-sagar/CAPSTONE

- *What are the resources used in the project? (libraries, APIs, databases, tools, etc)*

Resources

- Visual Studio Code
- Postman
  - testing CRUD operations
- Draw.io
  - user flow, architecture and database diagrams
- Figma
  - wireframes

Front-end

- React
- Vite
- Material UI/Icons
- Emotion
- Iconify
- Axios
- React Router Dom

Back-end

- Axios
- Bcrypt
- Cors
- Dotenv
- Multer
- Node.js
- Express
- MySQL
- Sequelize