Good afternoon TA and friends we are Meta-4. Today, our project showcase will be about our Lost and Found app.
Firstly, the purpose of our app is to provide a platform for Singaporeans to post their lost items and findings in order to facilitate the return of lost items to rightful owners. Hence, our target audience or users, rather, would be Singaporeans who are prone to losing their items.

Our mobile based application uses the MERN Stack with React Native to allow users to log into their profiles which will enable them to post lost or found items, as well as viewing those items and their details posted by other users.

Users are able to view a map with post marked on the map and clicking on the pins will navigate to the relevant post page.

Moving on to our functional requirements, for user registration and login, users must be able to create an account and login to the app to create or view posts. For Lost Item Post: Users can create a post about a lost item, including a title, description, picture, location, and contact information. They can also edit or delete their post as needed. For Found Item Post: Users can create a post about a found item, including a title, description, picture, location, and contact information. They can also edit or delete their post as needed. Next, for Search function: Users can search for lost or found items based on keywords, location, or category. Lastly, Map function: Users can view posts interactively based on their locations.

For non-functional requirements we have the following:

Security: The app must have appropriate security measures to protect user data and prevent unauthorized access.

Performance: The app must be able to handle a large number of posts and searches without experiencing significant delays or crashes.

User-friendly interface: The app should have an intuitive and easy-to-use interface that allows users to quickly create or view posts and search for lost or found items.

Constraints: The app must comply with relevant laws and regulations related to data privacy and user protection.

Assumptions: The app assumes that users will provide accurate and truthful information in their posts and that they will act in good faith when returning lost items

Dependencies: The app depends on users to create and search for lost or found item posts, as well as to communicate with each other to coordinate the return of lost items.

Acceptance Criteria: Users can create a post about a lost item or found item with all required information. Users can search for lost or found items based on keywords, location, or category.

As for the use of a dynamic database, we went with ONEMAP API, a government data API, to geocode the location of our post. The ONEMAP API provides us with the longitude and latitude of locations in Singapore using text matching for our location description. This enables us to link posts to a location pin which will enable us to display on our interactive map.

This here is our most updated dialogue map which includes all the primary and essential functions of our app. As you can see, there is only 1 end and start point.

Moving on, I will now be briefly showing you all our sequence diagrams for our use cases that are of high priority. We made use of the different classes and their logical sequences.

As for the various principles we used, we mainly target the SOLID principles. Under Single Responsibility, we made sure that for both the backend and frontend, each controller is responsible for a single task and performs it well. For Interface Segregation Principle, we made sure that the frontend side only called the relevant control classes and APIs. For Open/Closed, our code is open for extension of new functions, such as rewards system in the future, without having to edit the original code.

In terms of our test driven development, we separated into the backend and frontend for fullstack development. For the backend, we used test-cases for the getting and posting of our backend API and tested them with Postman using the appropriate variation of valid and invalid inputs. For the frontend, we made sure that user interface shows the appropriate invalid error screen when there is an invalid input. On top of that, we also ensured separation of concerns by ensuring that different parts of an application separated into distinct sections, each of which addresses a separate concern.  This also helps to keep the code modular, easier to maintain and test, and allows different developers to work on different parts of the codebase without interfering with each other's work.

Some challenges we faced included:

- Learning curve: Learning new languages, which leads to learning new syntax, libraries, and frameworks. Most of us are unfamiliar with the languages like react and node.

- Code quality: As we are new to MERN stack, specifically React-Native, this led to incoherent codes at times, spending hours debugging.

- DIfferent writing styles: Even when using the same language (JavaScript) across our stack there were also some difficulty understanding each other’s code due to the difference in the way of writing code

- Integration: Integrating code especially when we are new to the language can be challenging. The code may not integrate well with existing code, or it may introduce new bugs or issues.

- Limited resources: There may be limited resources available for learning the new language, such as documentation, forums, and libraries. This can make it difficult to troubleshoot issues or find solutions to problems.

To address these challenges, we have frequent meetings in order to communicate effectively, set clear expectations, and establish coding standards and best practices. We also allocate sufficient time for learning and experimentation and leverage available resources to accelerate the learning process. This helps everyone to be clear and focus on their own work and make us

an effective team. Results in speedy integration of code and delegation of workload to make the process smoother

Some benefits of our app includes:

- Streamline the lost&found process for users
- Intuitive and easy to use for users
- Functional search filters and map to ease searching of posts
- Features a visually stunning design that is both modern and functional
- Real-time updates of objects

Some future developments and improvements we may intend to take up include:

- Rewards System— To encourage desirable behaviours. Can be designed as a modular component that can be easily added to our app

- Chat System — To allow ease of communications among users to facilitate returns. Can be designed as a modular component that can be easily added to our app

Now i will be demonstrating our application.
Firstly i will register an account using

{name: test
email: test@test.com
password: test
phone: 12345678}

Now i will show what happen when we register a new account with the same email.

{name: test2
email: test@test.com
password: test2
phone: 12345678}

As you can see, sign up is unsuccessful as email has to be unique.

Now I will be showing our login function. For login, we require email and password field. Invalid email and password will result in unsuccessful login. For password we also used bcrypt hash function. Our authorization uses Json webtoken to authenticate.

Now i will be showing creation of post whereby I can choose to post a found/lost post depending on the context.
As you can see, our post created will show up on the home page, as well as under the user profile and in the maps.

I can also edit my post by clicking on my own post and the edit button.
I can also choose to delete my post by click on the delete button.

Also when I lost an item and I see other users posting that they found an item , I can make a request to the post owner, by clicking on the “I found it”

So after I send a request, I will log in to the post owner account, and view the activity page. As you can see there is a new request made, and I will approve the request.  Now the post will be counted as resolved and we will no longer see it in our homepage. Under the user profile, it will also be shown under resolved page. The maps will also not show the posts anymore as it has been resolved.

.