Stage 6. Presenting the Final Solution

Student: Merab Kardava

Project: University Lost & Found item tracker

What kind of app is it? A web application built with Spring Boot where students and staff can report lost or found items on campus.

Who is this project for? University students, teachers, and campus staff.

What needs will it satisfy? It helps users track lost belongings, see if someone found their item, and connect with the person who found it. It reduces time and confusion in recovering lost items.

Major features are reporting of lost or found items, report editing and claiming and notification.

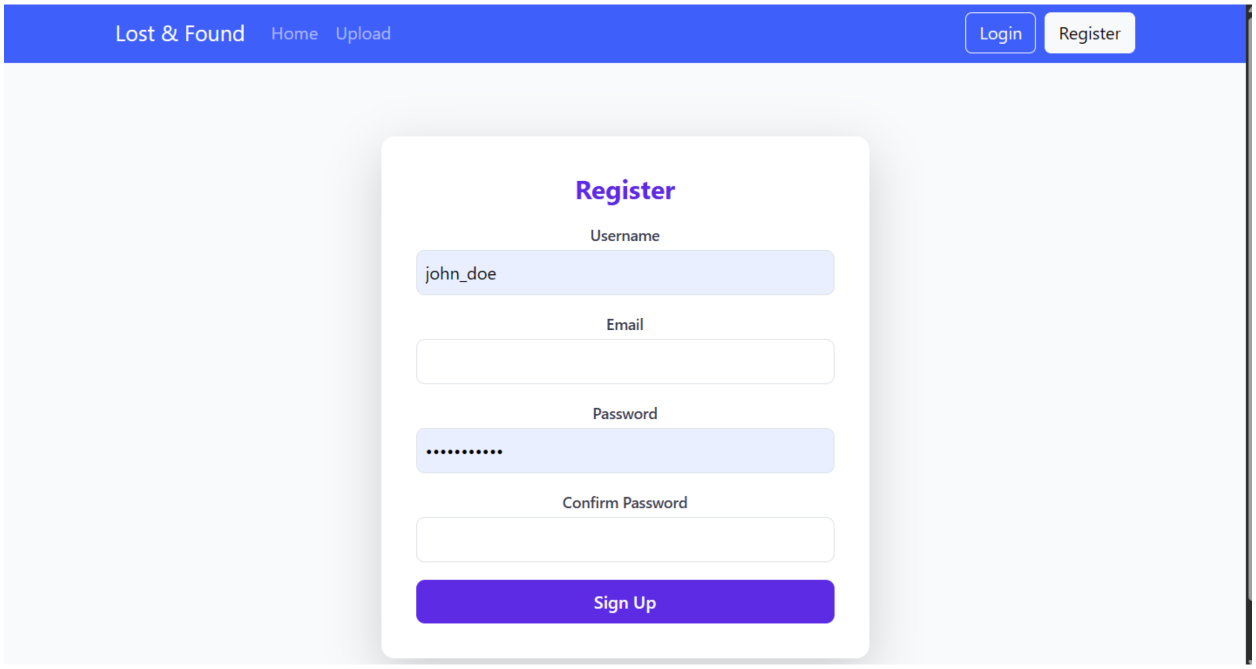
Technologies used – SpringBoot, Spring Security, Spring Data, SpringMVC , Lombok(for reduced boilerplate), H2 in memory database, React for front end and it's various libraries: Axios, React bootstrap, React router DOM, tailwind (for quick styling).

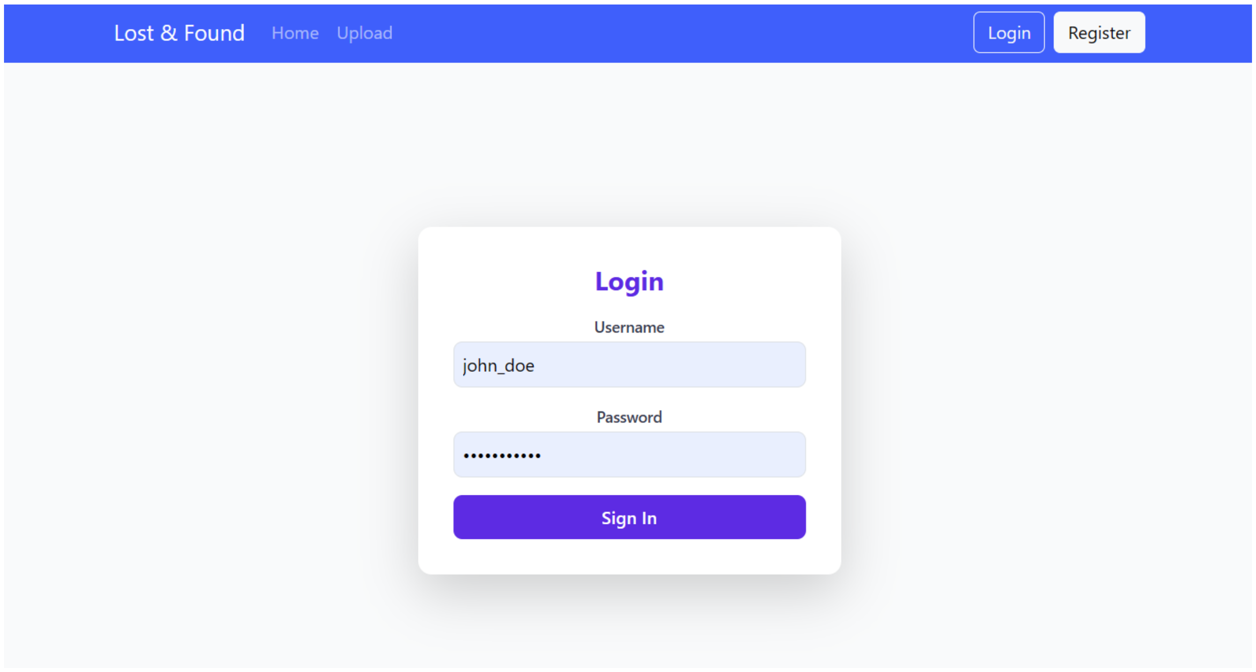
Features

Registration and login

Registration and login pages are represented by register and login react pages in the front end that and use functions in auth api service to interract with rest api in the back end.

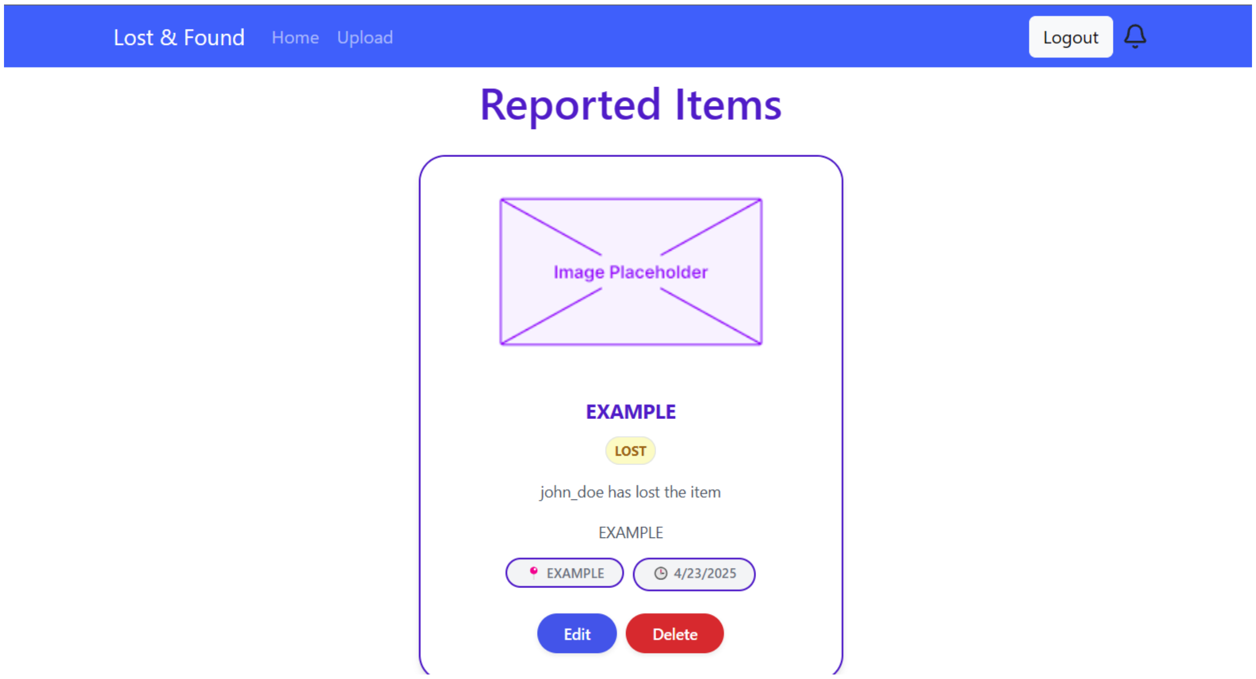
Authentication is cookie based.



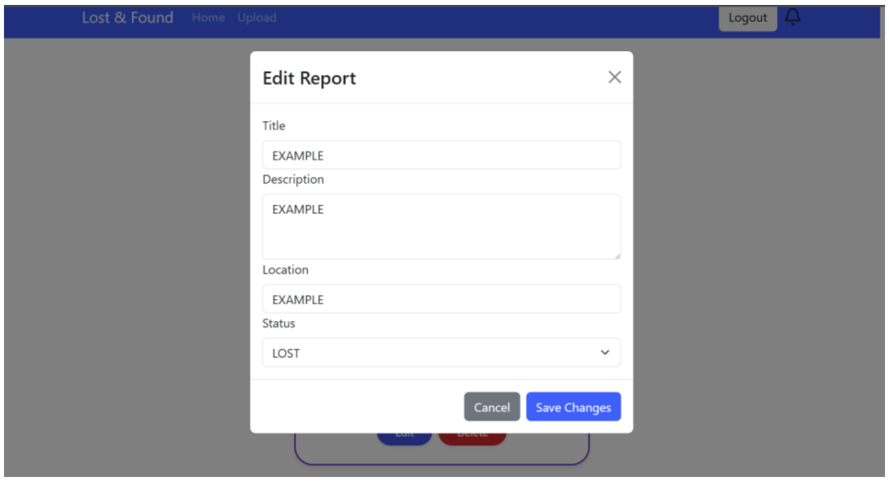


Looking at reports

ReportPage makes a get request to back end and receives an array of reports that is then used to create ReportComponents

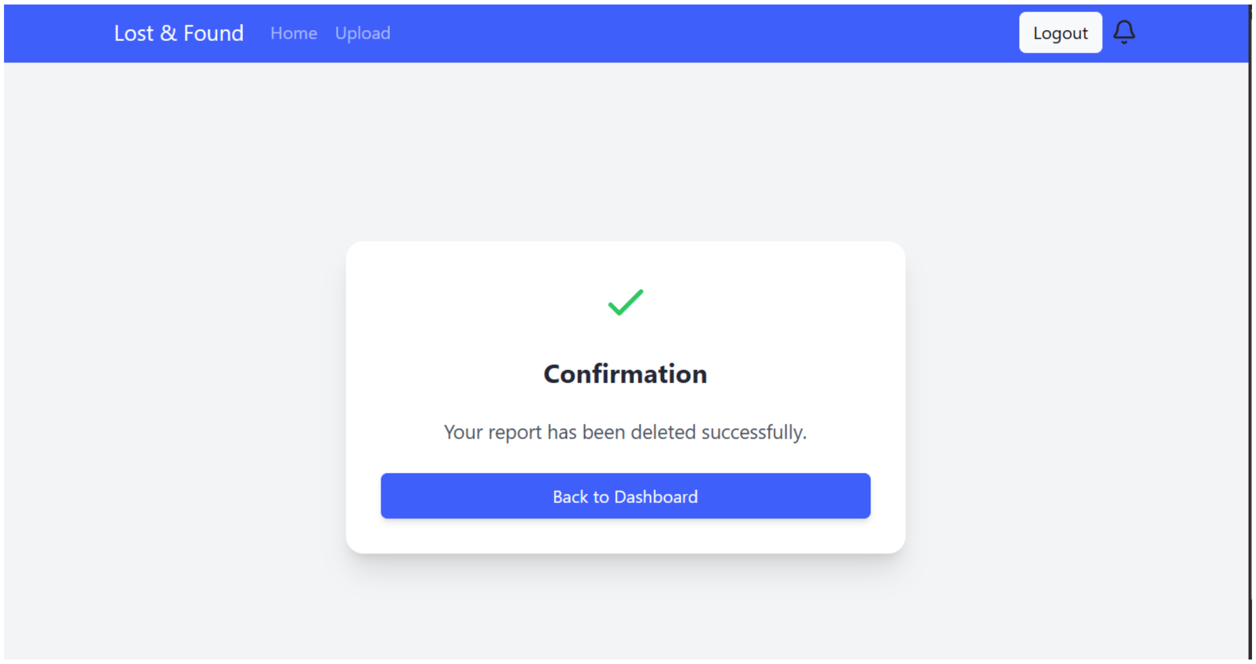


Report component itself consists of a image and information like location creation time and status LOST or FOUND. It also comes with different buttons depending on status or user . If the report is created by the current user you have the option to edit or delete the report. Clicking Edit button opens a modal that lets you update the report then calls a put request. User information is stored in session storage .



Title
EXAMPLE
Description
EXAMPLE
Location
EXAMPLE
Status
LOST

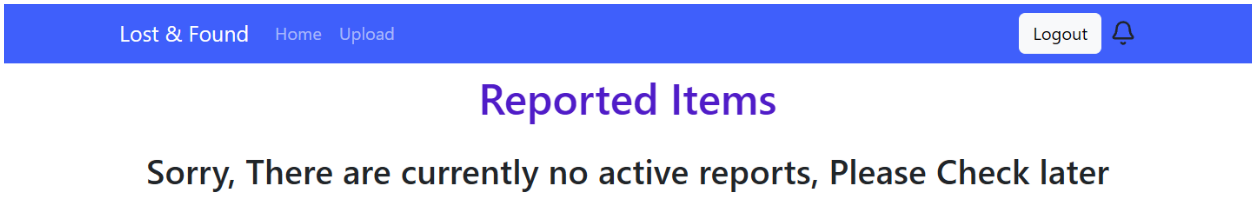
Pressing delete calls a function that makes a delete request and opens a confirmation page



Confirmation

Your report has been deleted successfully.

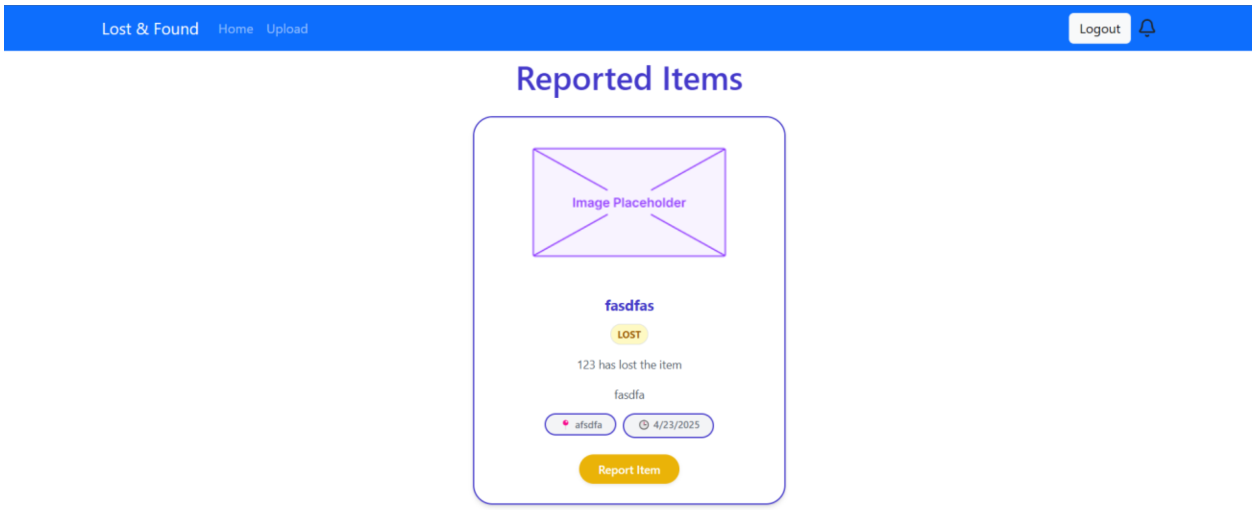
If there are no



Reported Items

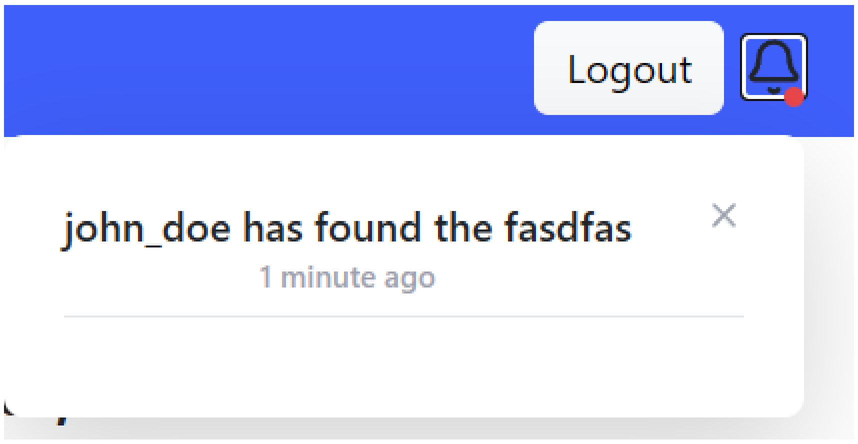
Sorry, There are currently no active reports, Please Check later

You can claim items found by other users as yours or report that you found the item shown in the report by clicking the button. Claming/reporting the item or deleting the report opens a confirmation page and sends a notification to the author of the report.



Notification component calls a get request and receives current user's notifications which are then displayed . you can see a notification messages here, see how long ago the report was claimed/found or delete the notification.

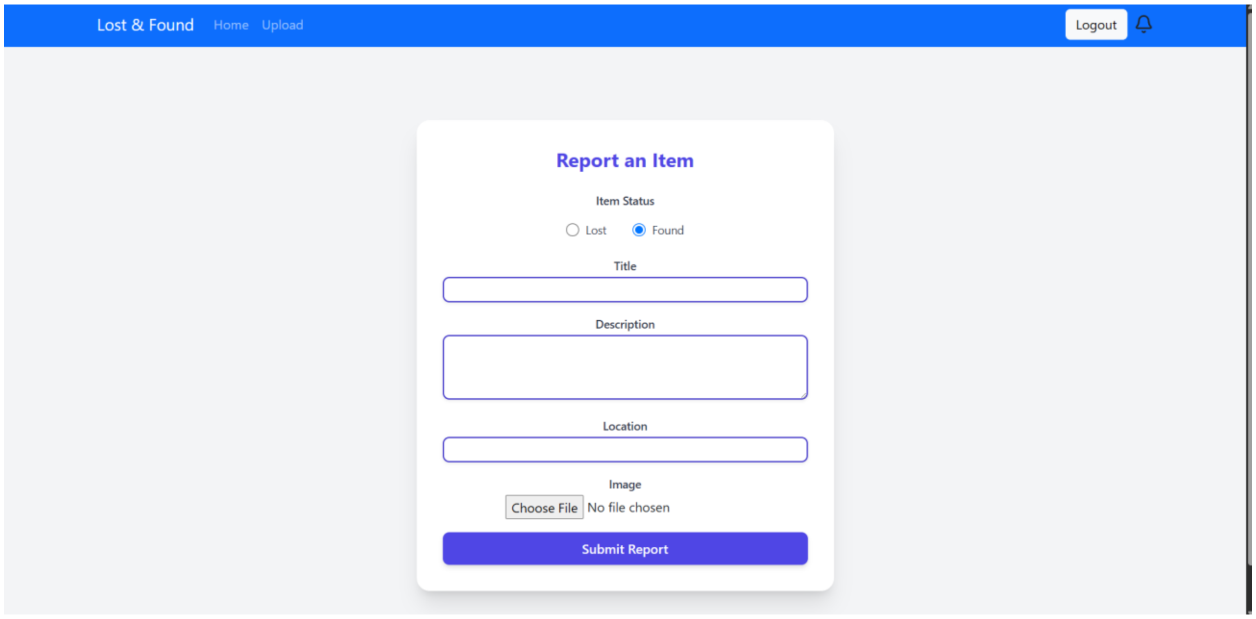
Notifications are updated every 10 seconds.



john_doe has found the fasdfas
1 minute ago

Report making.

You can make reports with this form. After Submission you are redirected to a confirmation page and a post request is called to add the report to the database. In the database images are stored only as a path. Actual images are stored in the back end uploads folder and accesible when authorized.



Actions like logging in report creation or deletion are stored in the database with a time stamp and username of subject for auditing purposes. This is handled by AuditLogService in the back end and represented by an AuditLog Entity.

Code Structure

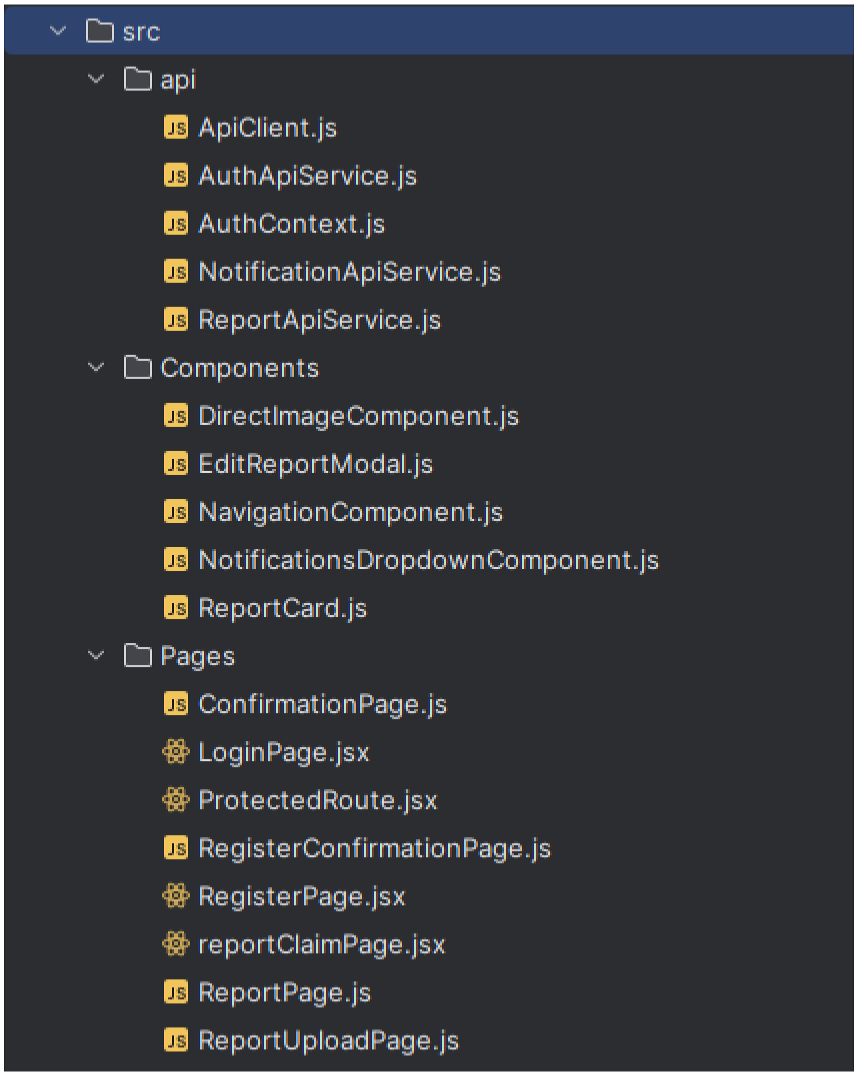
In front end code is mainly divided into Pages, Components and api.

Pages file holds react functional component of the website's pages.

Components file holds various components that make up the pages.

Api file holds various methods that are used to communicate with the rest api and configruations.

It also contains an AuthContext that is used for app wide state sharing and tracking of the authentication status and user information.



```
src
  api
    ApiClient.js
    AuthApiService.js
    AuthContext.js
    NotificationApiService.js
    ReportApiService.js
  Components
    DirectImageComponent.js
    EditReportModal.js
    NavigationComponent.js
    NotificationsDropdownComponent.js
    ReportCard.js
  Pages
    ConfirmationPage.js
    LoginPage.jsx
    ProtectedRoute.jsx
    RegisterConfirmationPage.js
    RegisterPage.jsx
    reportClaimPage.jsx
    ReportPage.js
    ReportUploadPage.js
```

In the back end code is divided into
Config contains configruation beans for security/cors and static image sharing.

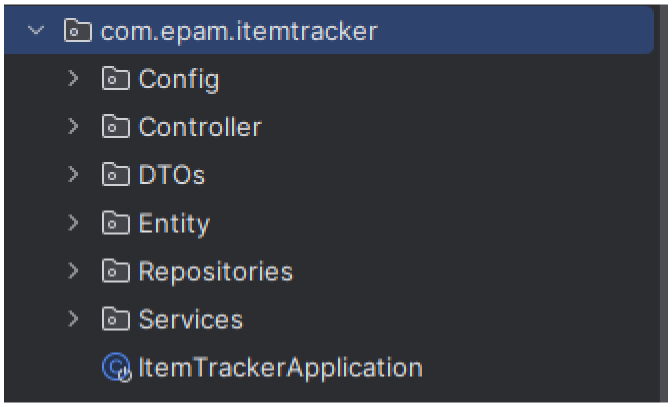
Controller folder contains rest controllers for rest endpoints.

DTOs contains data transfer objects.

Entity folder contains table entities and enums that are used in these entities.

Repository folder holds Repositories for the entities

And service folder holds services.


```
com.epam.itemtracker
    Config
    Controller
    DTOs
    Entity
    Repositories
    Services
    ItemTrackerApplication
```

Conclusion

In conclusion this app makes looking or reporting lost items easier for users. Has a simple and intuitive design and is user friendly.