



CS319 Object-Oriented Software Engineering D6 Report

Süleyman Yağız Başaran

22103782

Gülbera Tekin

22003354

Onur Tanınmış

22003312

Serhat Yılmaz

22002537

Melisa İrem Akel

22003923

Betül Doğrul

22003559

Table of Contents

CS319 Object-Oriented Software Engineering D6 Report	1
1. Introduction	2
2. Lessons Learned	3
3. User-Manual	3
3.1 User Guide	3
3.2 High-Level Description	17
3.3 How to Install the Software	17
3.4 How to Run the Software	17
3.5 How to Report a Bug	17
3.6 Known Bugs	18
3.6.1 Donation, Borrow, SHS Item Share is not working	18
3.6.2 Comment liking is not working	18
3.6.3 Freezone Share User Profile Picture is buggy	18
4. Developer-Manual	18
4.1 How to Obtain the Source Code	19
4.2 The Layout of Our Directory Structure	19
4.3 How to Build the Software	20
4.3.1. Backend Instructions	20
4.3.2. Frontend Instructions	20
5. Work Allocations	21

1. Introduction

This is the final report of our CampusConnect project. The application is for the use of Bilkent members, namely the students, teachers, and alums. Customer expectations were highly considered during the implementation process of the application's features. Every user will sign up with their Bilkent e-mail, name, surname, username, and password by clicking the "Sign Up" button and log in using their Bilkent e-mail or username and password by clicking the "Login" button on the welcome page of the application.

After logging in, the users can see their profile page, the FreeZone feed, and the Lost & Found, Second-Hand Sales, Borrow, and Donations pages. They can edit their profile page and add and delete posts into the L&F, SHS, Borrow, and Donations pages and the FreeZone page of their account. They can report a lost ID card by only entering the person's Bilkent ID into the "ID Card Report" part, and the person whose ID has been reported will be notified immediately by a notification. There is also a feature called "Live Chat," which allows users to text each other in a joint group called Chatroom and also chat privately. Also, a button called "Bug Report" will be available for users to report

a problem to us. There is also an extra part on the right edge of the application feed where the users can read different news related to Bilkent and computer science.

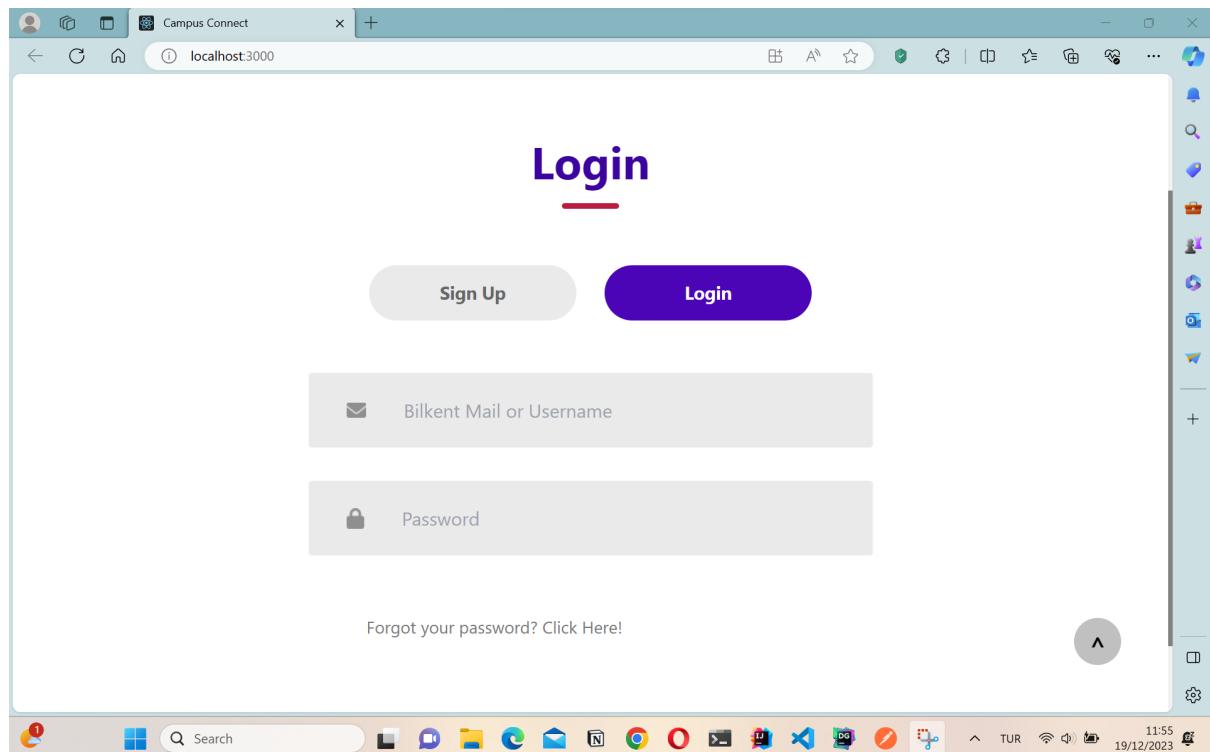
2. Lessons Learned

The group members needed to familiarize themselves with the technologies utilized during the coding. The learning curve took longer than we expected. However, all of us learned new technologies during the project. Moreover, we realized that having a shared database is very beneficial while connecting the database with the frontend. We also learned that little bugs can be hard to fix by directly coding, analyzing the code, and then trying to fix the bug faster. However, it can take longer.

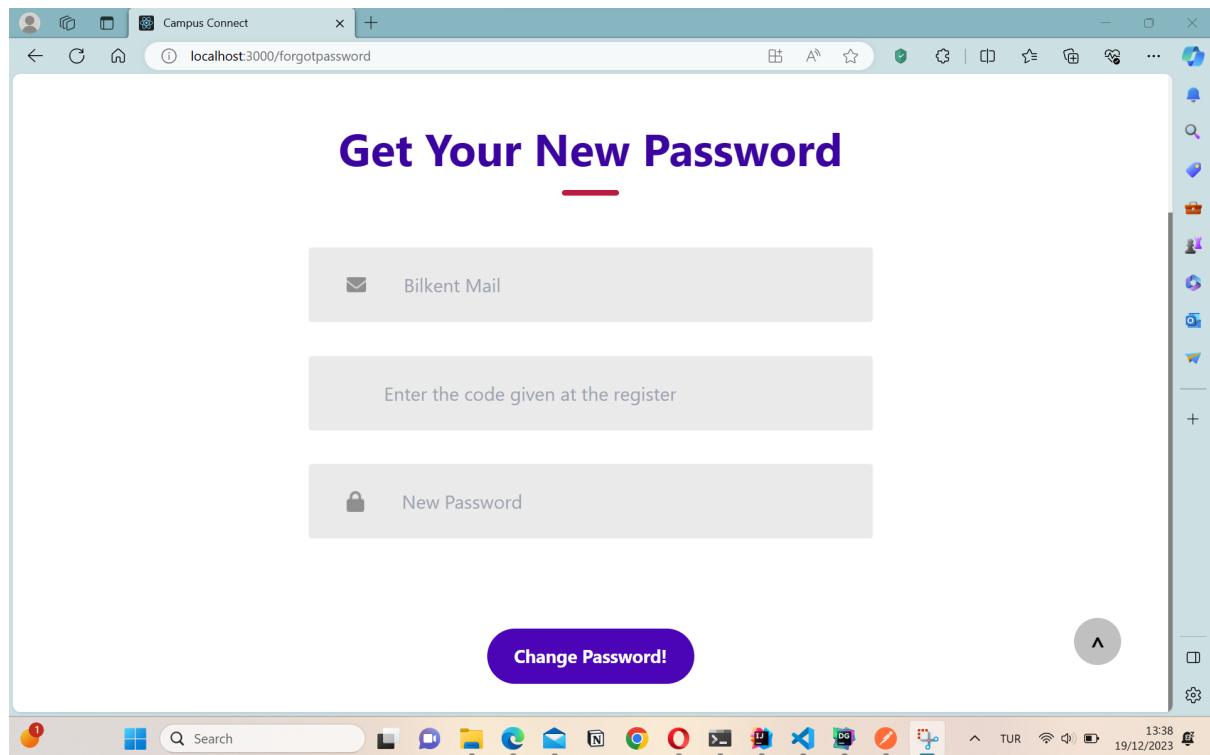
3. User-Manual

3.1 User Guide

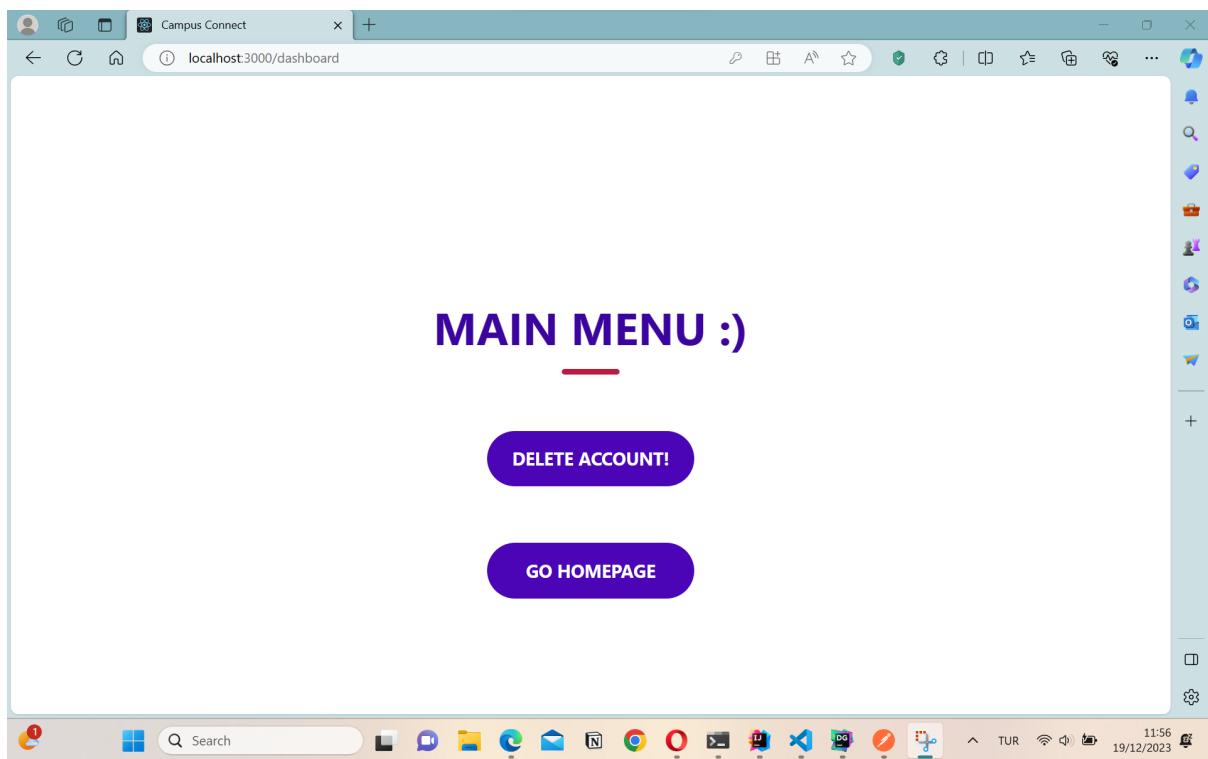
When the user opens the application, the following page appears on the screen. Users can register, log in, or push the forget password button.



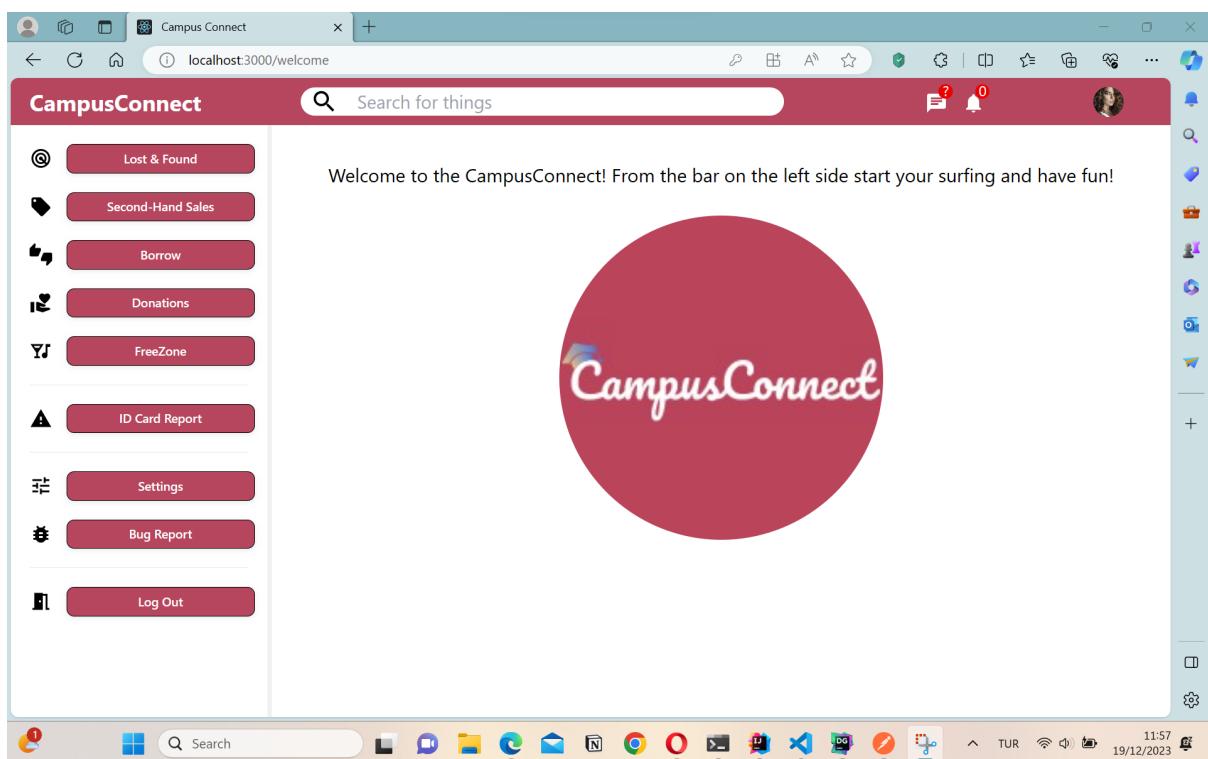
When the forget password button is pushed, the following page appears, and the code is assigned to every user and stored in the database.



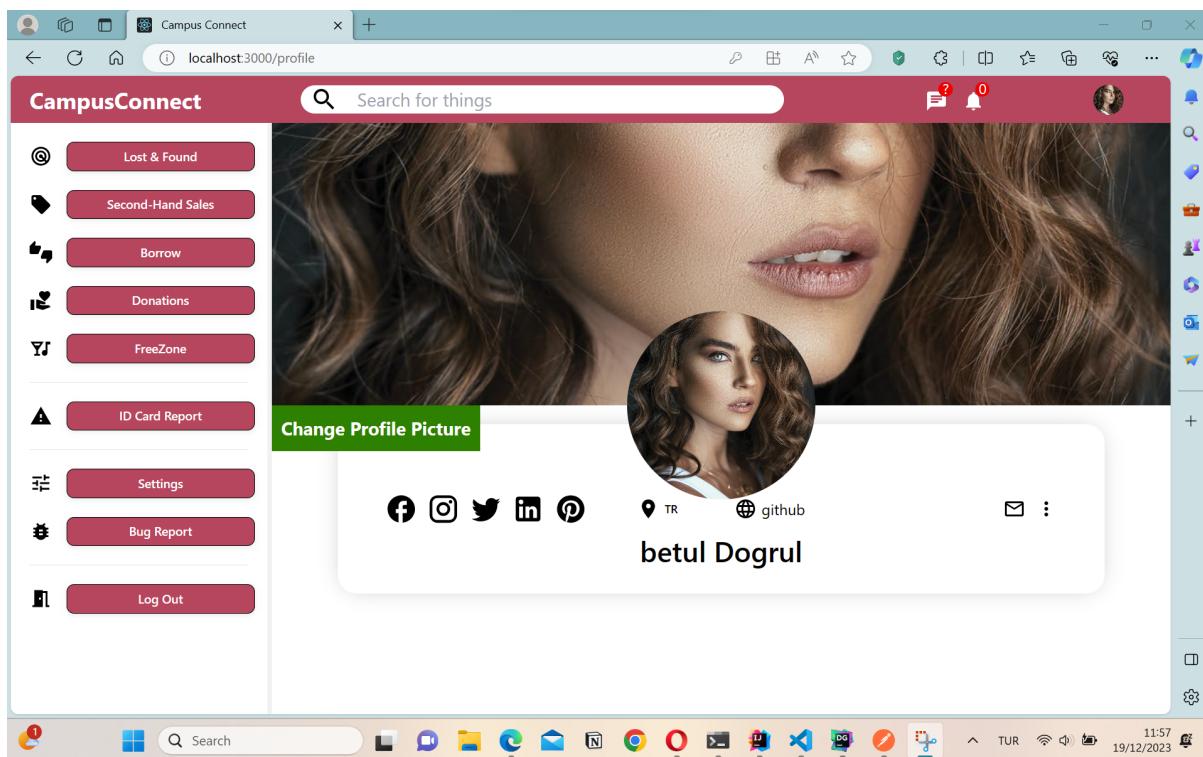
When the user logs in, the following page appears, and the user can delete their account or go to Freezone to scroll.



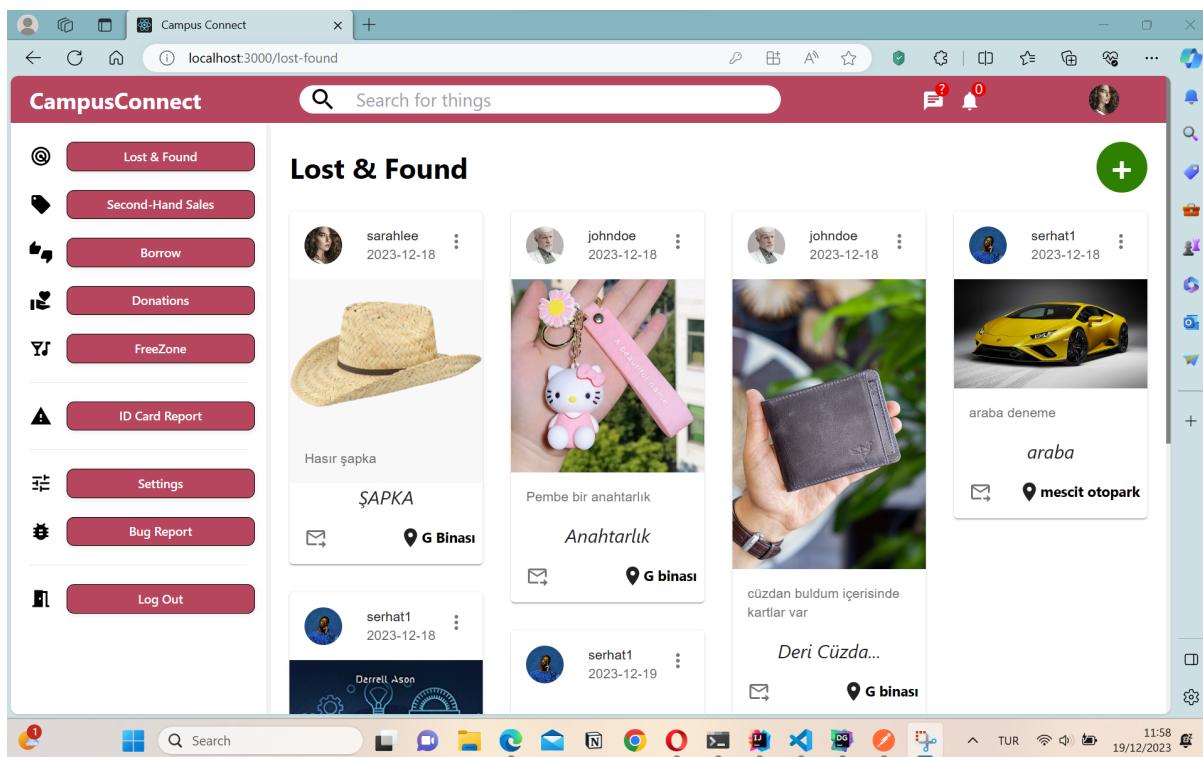
The following page appears if the user goes to the main page.



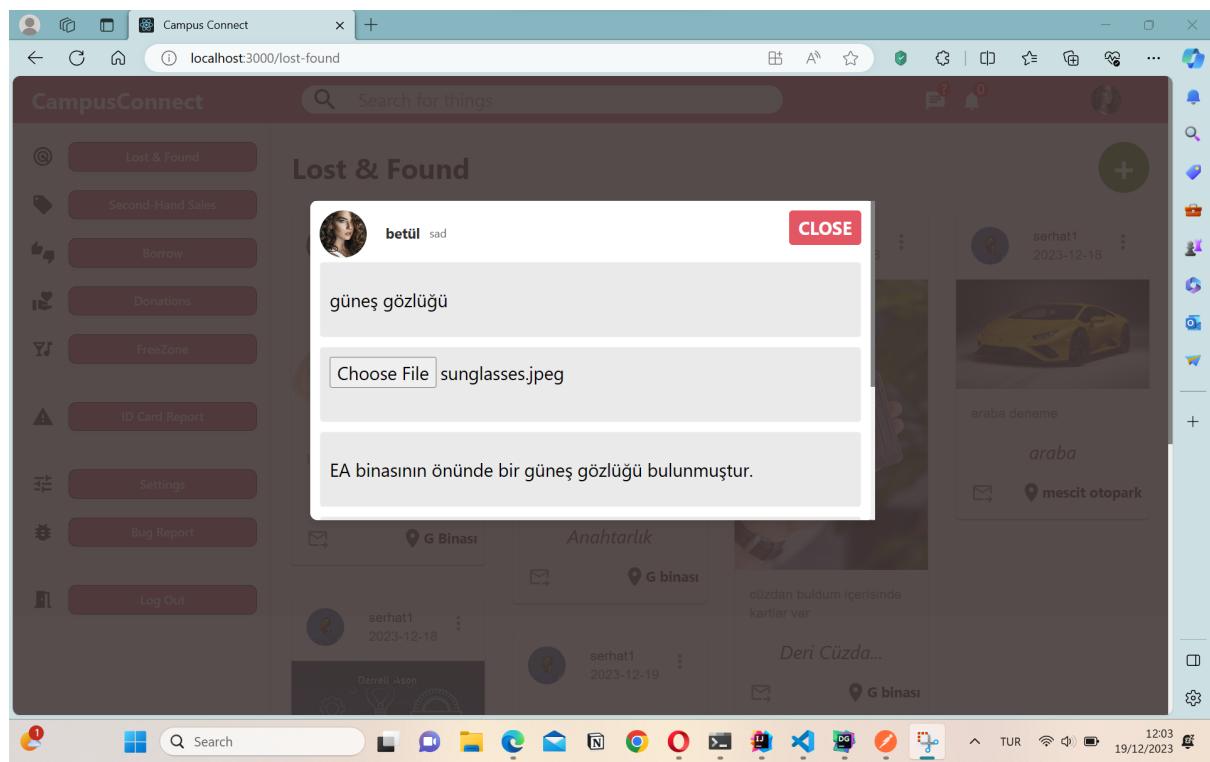
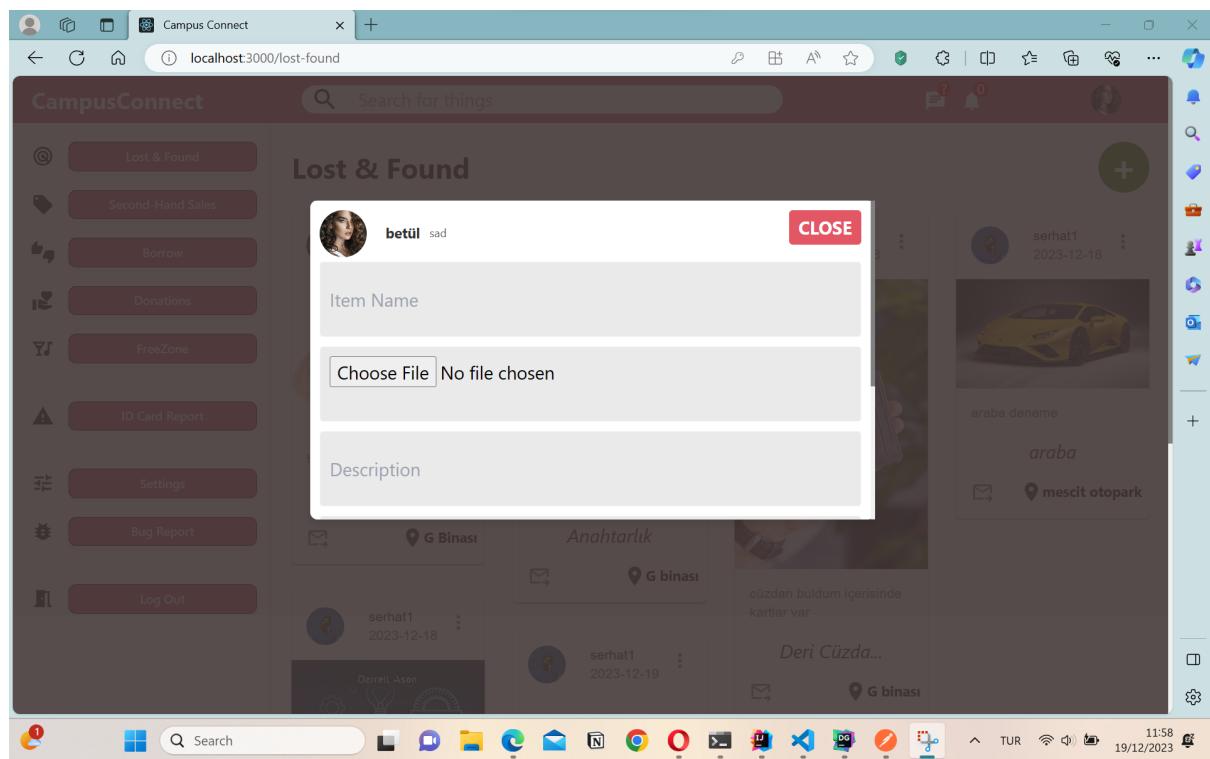
If the user clicks the right top profile photo, the user's profile page appears on the screen, as shown below.



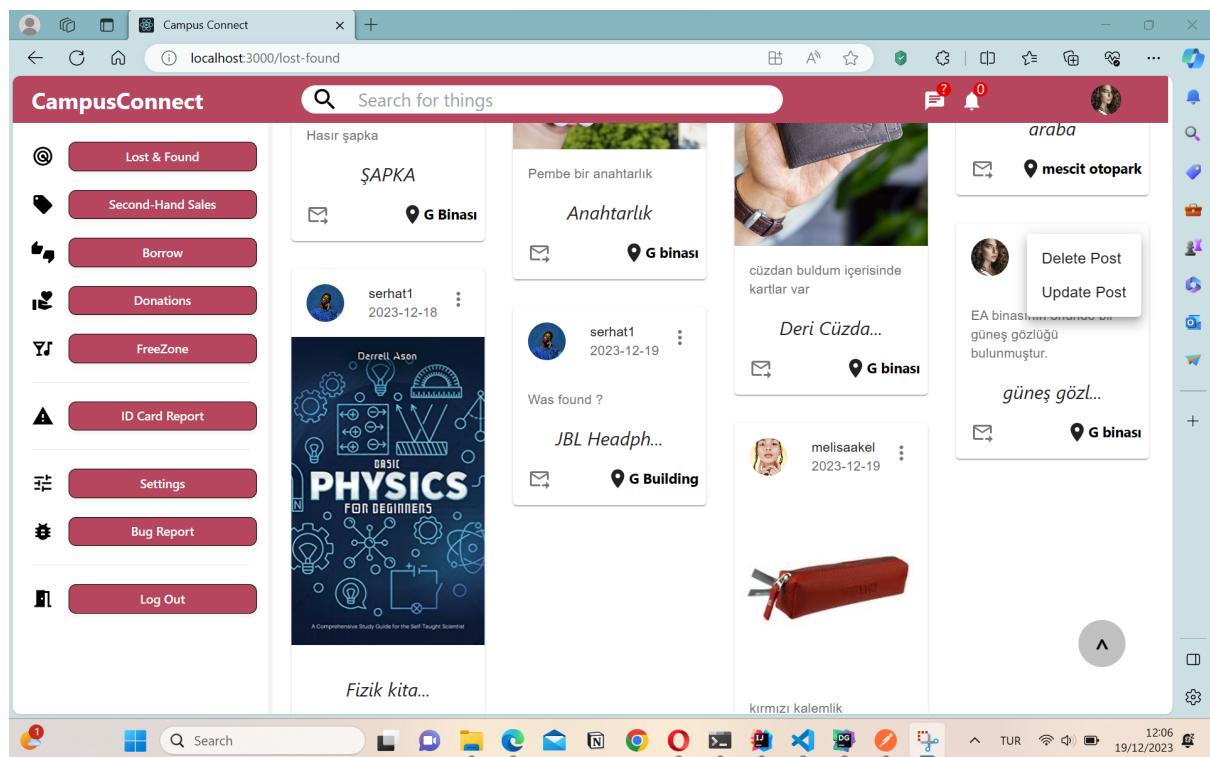
If the user clicks on the lost & found button on the left, the following page appears, and the user can add a post, delete their post, or chat with the person who posted the item.



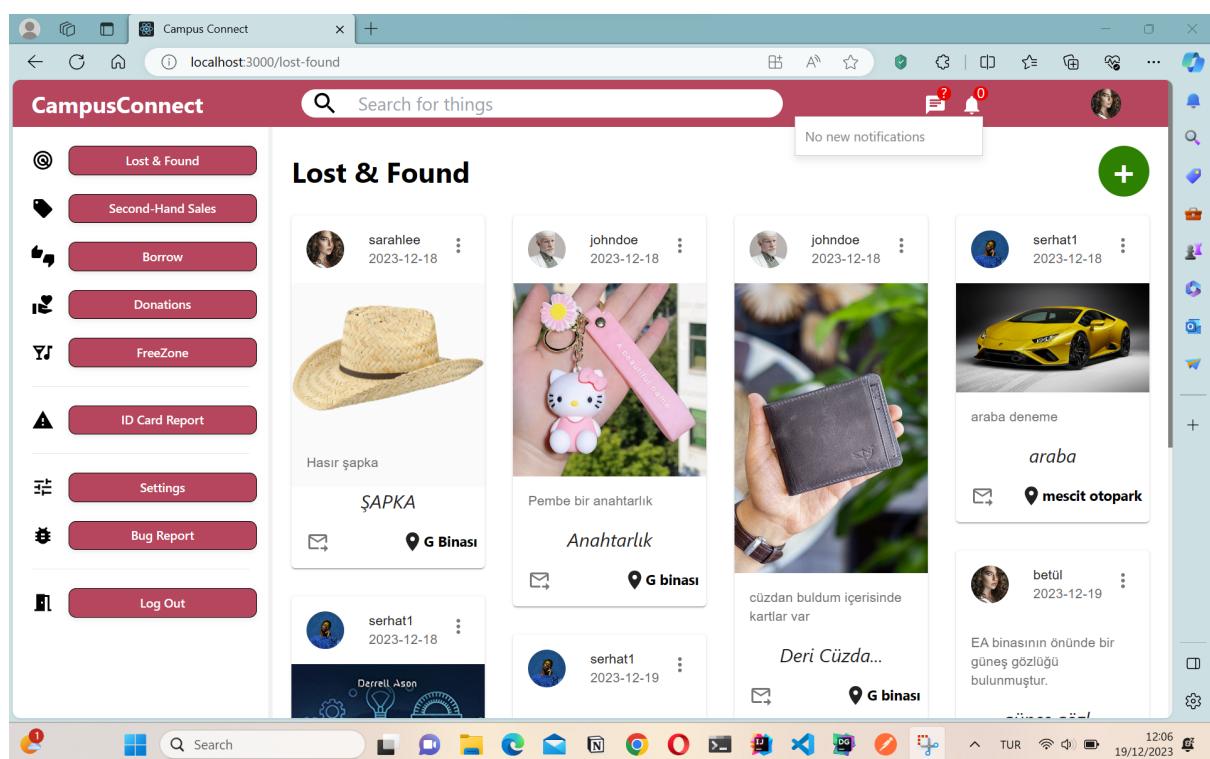
When the + button is pushed:



After the addition,



We also have a notification feature. It is instantiated when another user finds the user's ID card.



The page when the second-hand sales button is pushed,

Campus Connect

localhost:3000/second-hand

CampusConnect

Second-Hand Sales

johndoe 2023-12-18

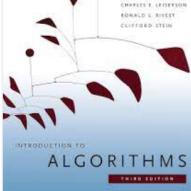


This pencilcase is in blue color and I dont use it anymore.

pencilcase

3 years old 50TL

serhat 2023-12-19



yepenyi, çizik yok

Introduction to ALGORITHMS

3 years old 750TL

serhat1 2023-12-19



Object-Oriented Software Engineering

3 years old 450TL

serhat1 2023-12-19



sadece 6 labda tertemiz kullanıldı

BASYS3

1 years old 6500TL

Search

12:08 19/12/2023

The page when the borrowing button is pushed,

Campus Connect

localhost:3000/borrows

CampusConnect

Borrowing

betül 2023-12-18



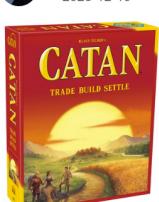
Gilgamış destanı

KİTAP

2023-12-18

2024-01-18

dzgdgz 2023-12-19



Catan oyunu

Catan

2023-12-19

2024-01-19

sarahlee 2023-12-19



yeni gibi

Satranç

2023-12-19

2024-01-19

sarahlee 2023-12-19



assets\post\basys3.jpeg

BASYS3

2023-12-18

2024-01-19

Search

12:08 19/12/2023

The page when the donations button is clicked,

Campus Connect

localhost:3000/donations

CampusConnect

Search for things

Donations

- johndoe 2023-12-18
- dzgdzg 2023-12-19
- dzgdzg 2023-12-19
- betül 2023-12-19

Category	Description	Author	Date	Action
Hand Book of COMPUTER SCIENCE	R. Gupta's Hand Book of COMPUTER SCIENCE	johndoe	2023-12-18	
Social Psychology	Social Psychology	dzgdzg	2023-12-19	
Concepts of Programming Languages	Concepts of Programming Languages	dzgdzg	2023-12-19	
Digital Design and Computer Architecture	Digital Design and Computer Architecture	betül	2023-12-19	

Lost & Found

Second-Hand Sales

Borrow

Donations

FreeZone

ID Card Report

Settings

Bug Report

Log Out

Search

12:08 19/12/2023

And the Freezone page:

Campus Connect

localhost:3000/freezone

CampusConnect

Search for things

What's in your mind?

dzgdzg

Günaydinn

Photo/Video #Tag Feeling Share!

Computer Science News

Amazon, Microsoft, and Google Help Teachers Incorporate AI Into CS Education

Long-time Slashdot reader theodp writes: Earlier this month, Amazon came under fire as the Los Angeles Times reported on a leaked confidential document that "reveals an extensive public relations strategy by Amazon to donate to community groups, school distri..."

Lost & Found

Second-Hand Sales

Borrow

Donations

FreeZone

ID Card Report

Settings

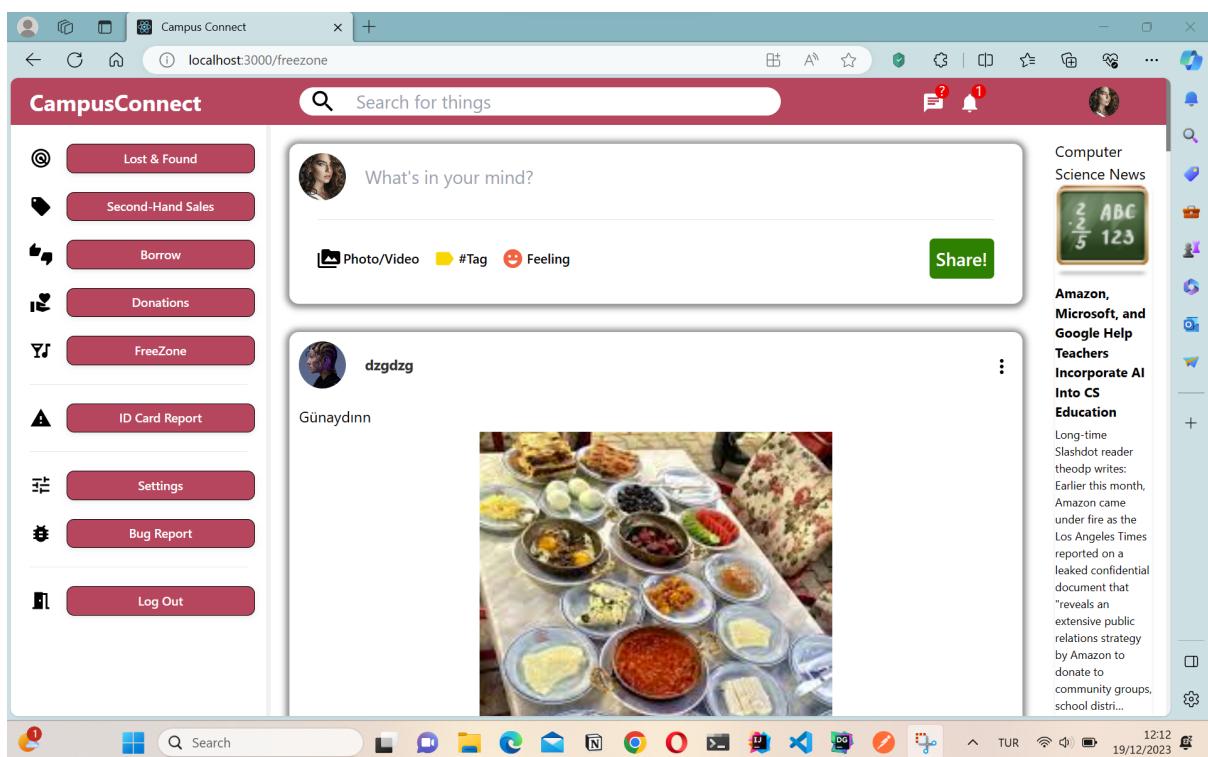
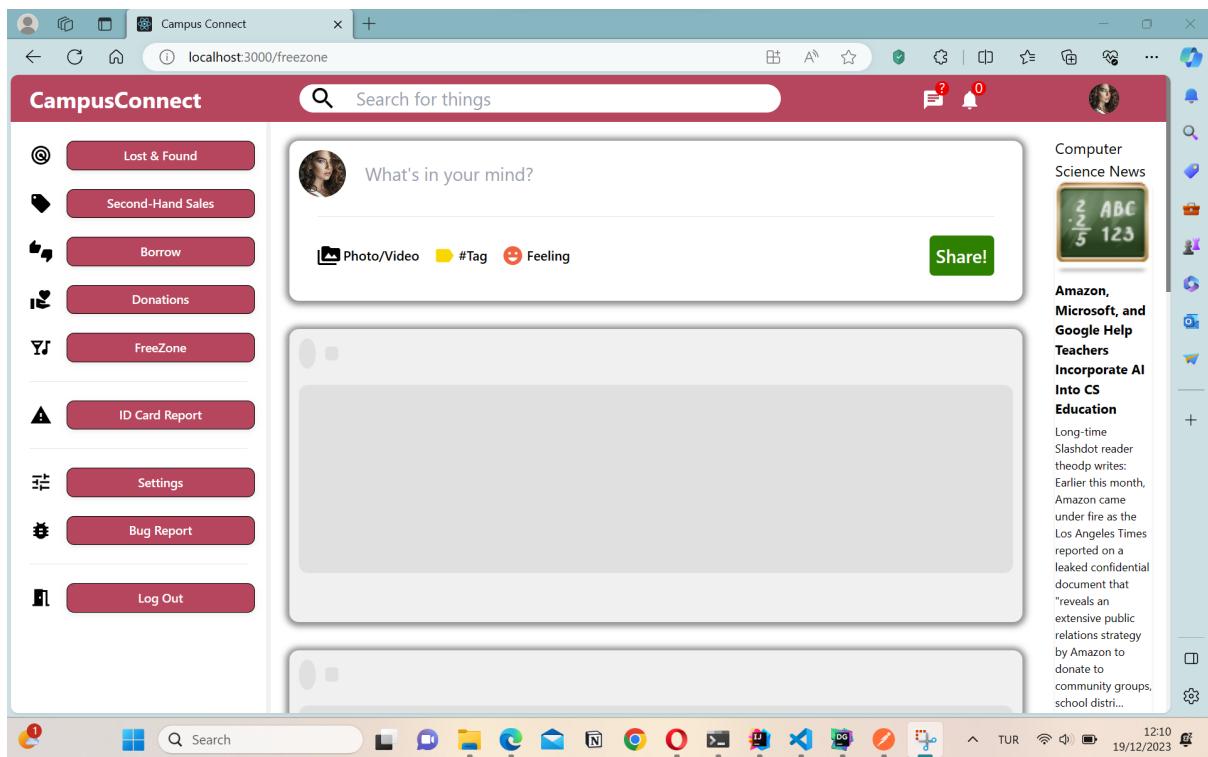
Bug Report

Log Out

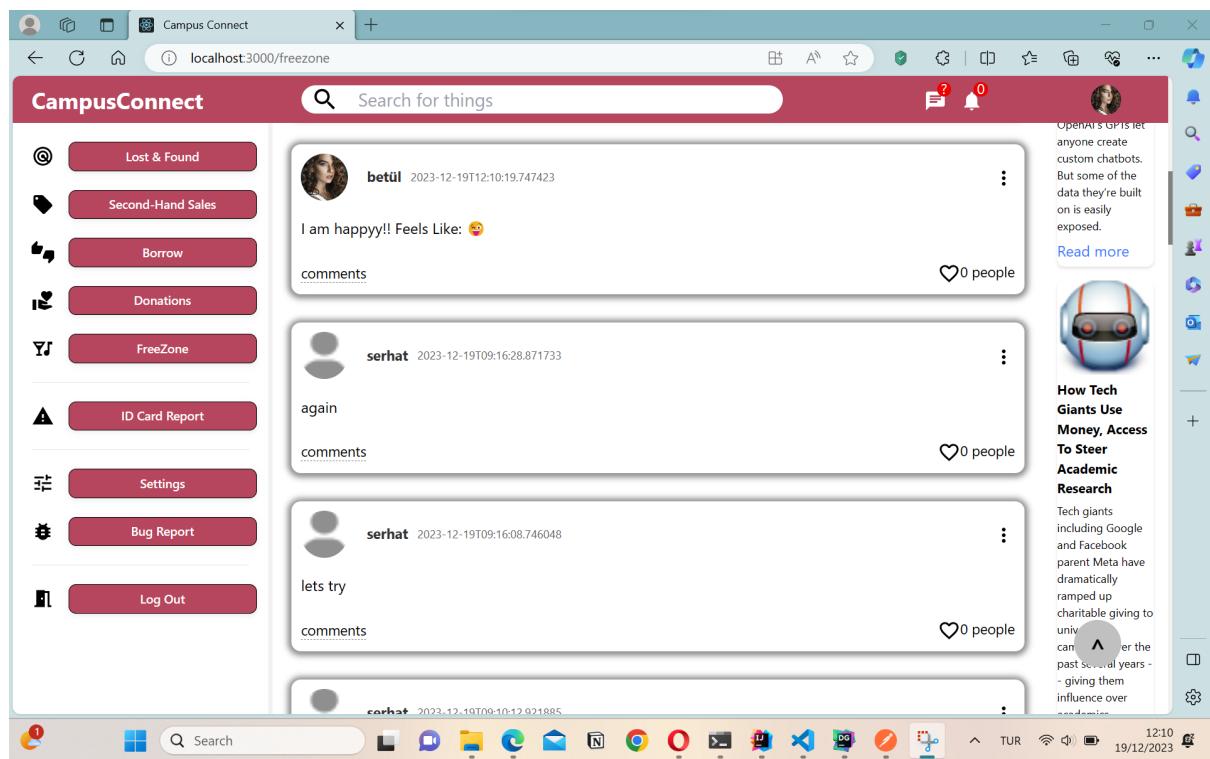
Search

12:09 19/12/2023

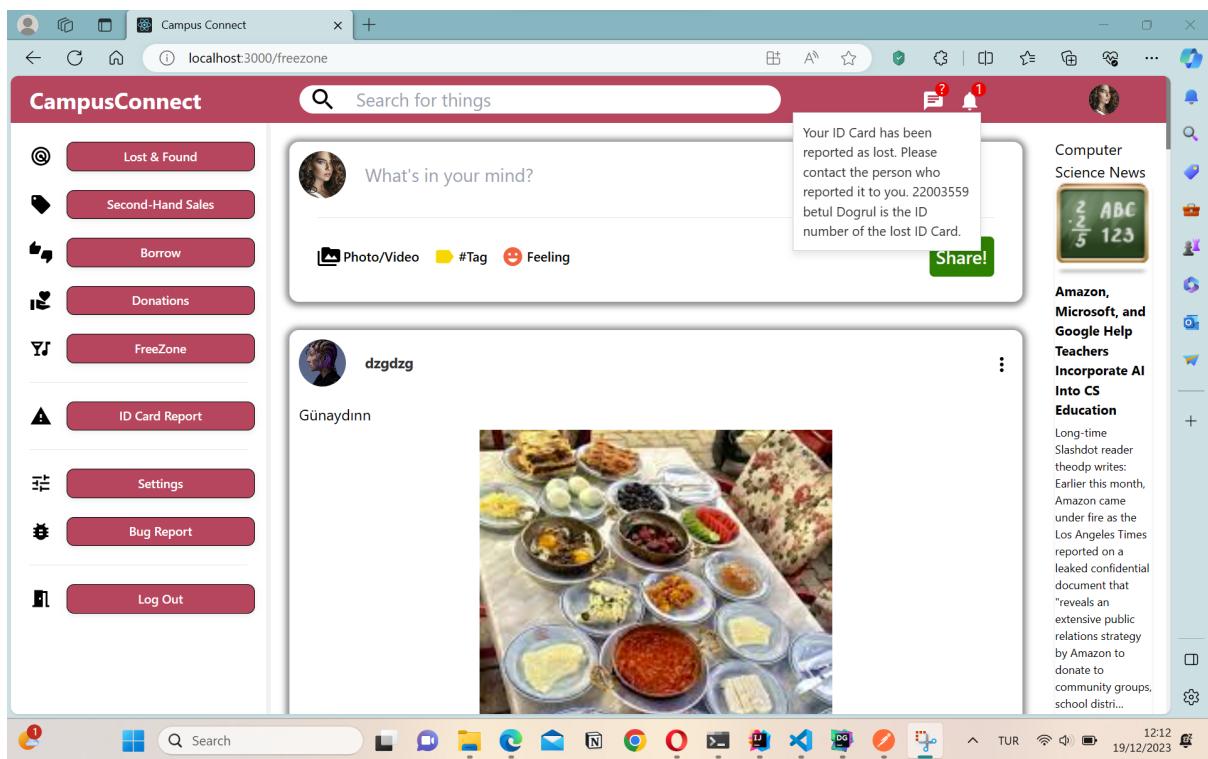
A post skeleton is shown for good use when loading the page.



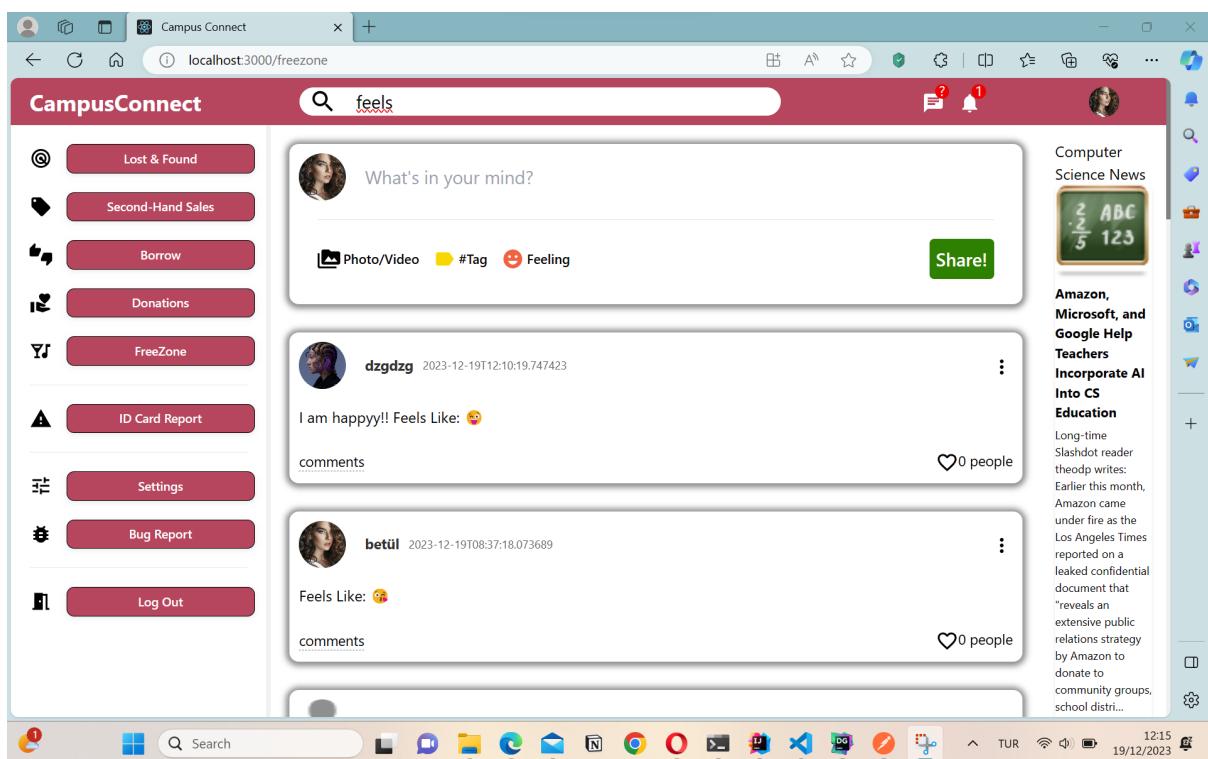
We have infinite scroll, and the posts are fetched by pagination for a good user experience.



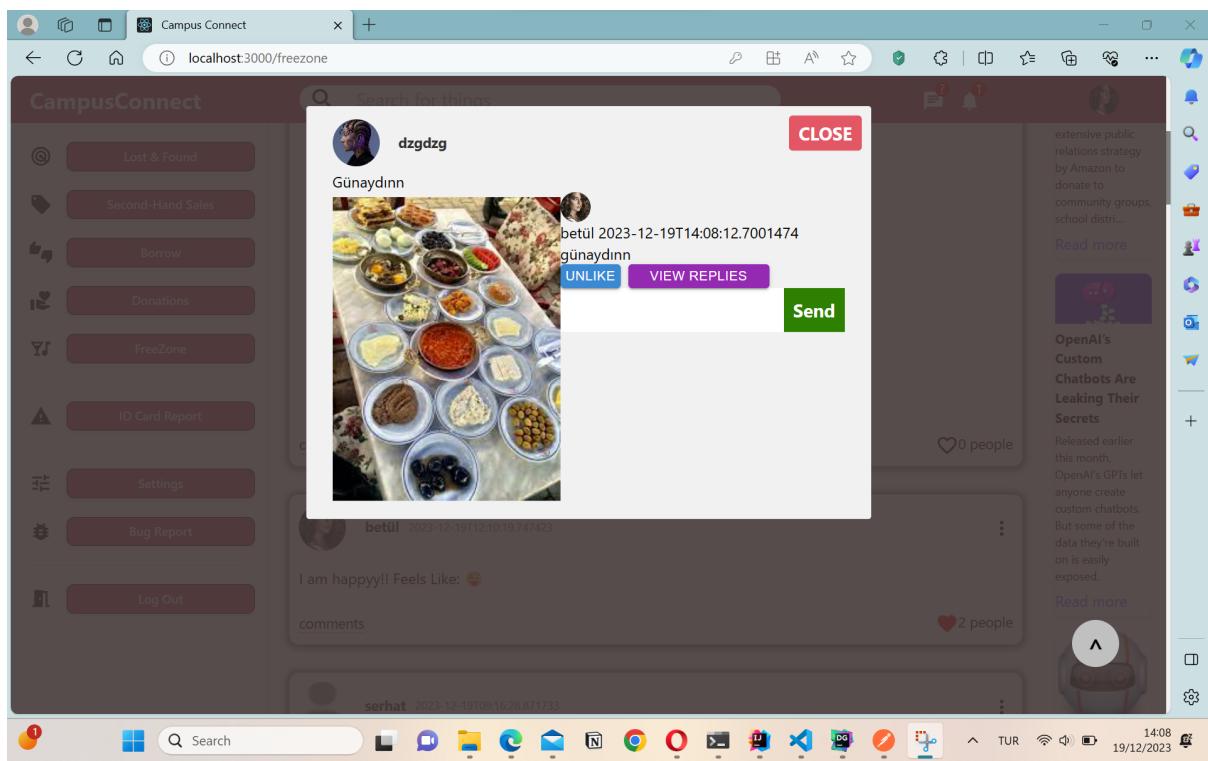
Here, you can see the notification when another user finds my lost ID somewhere, and it comes as a notification.



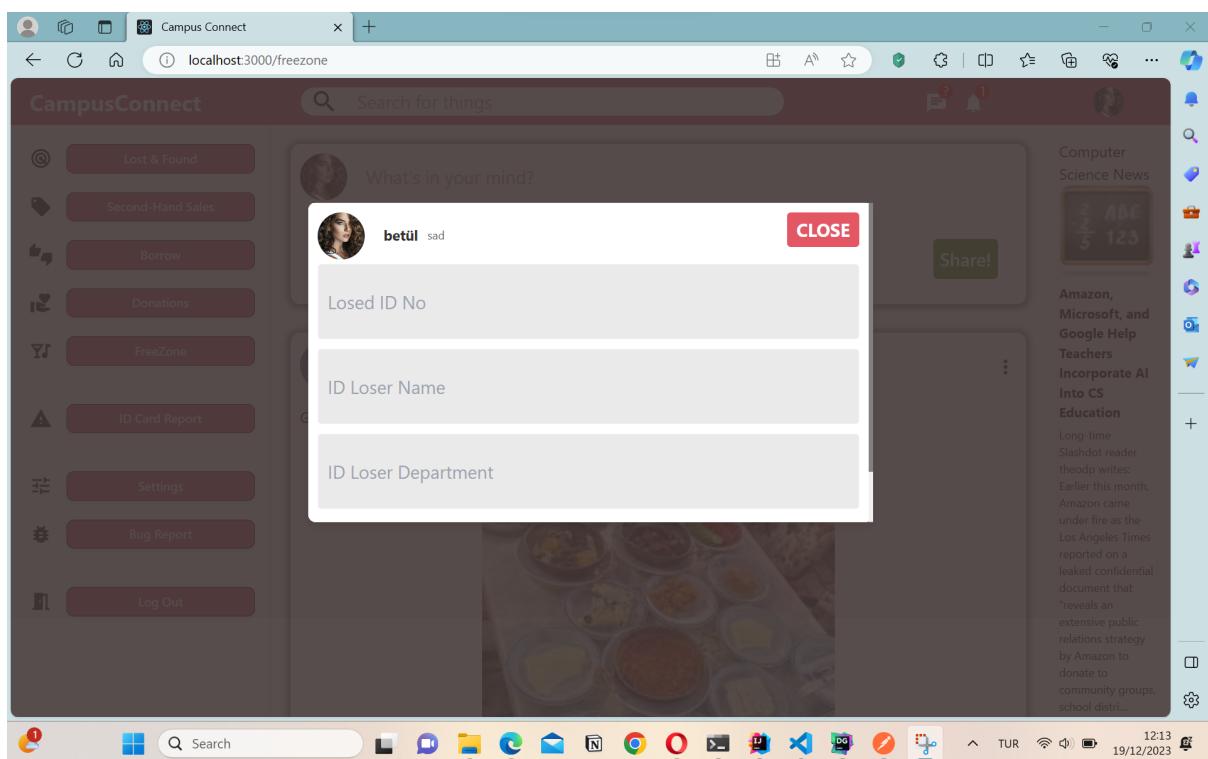
The user can also search for an item.



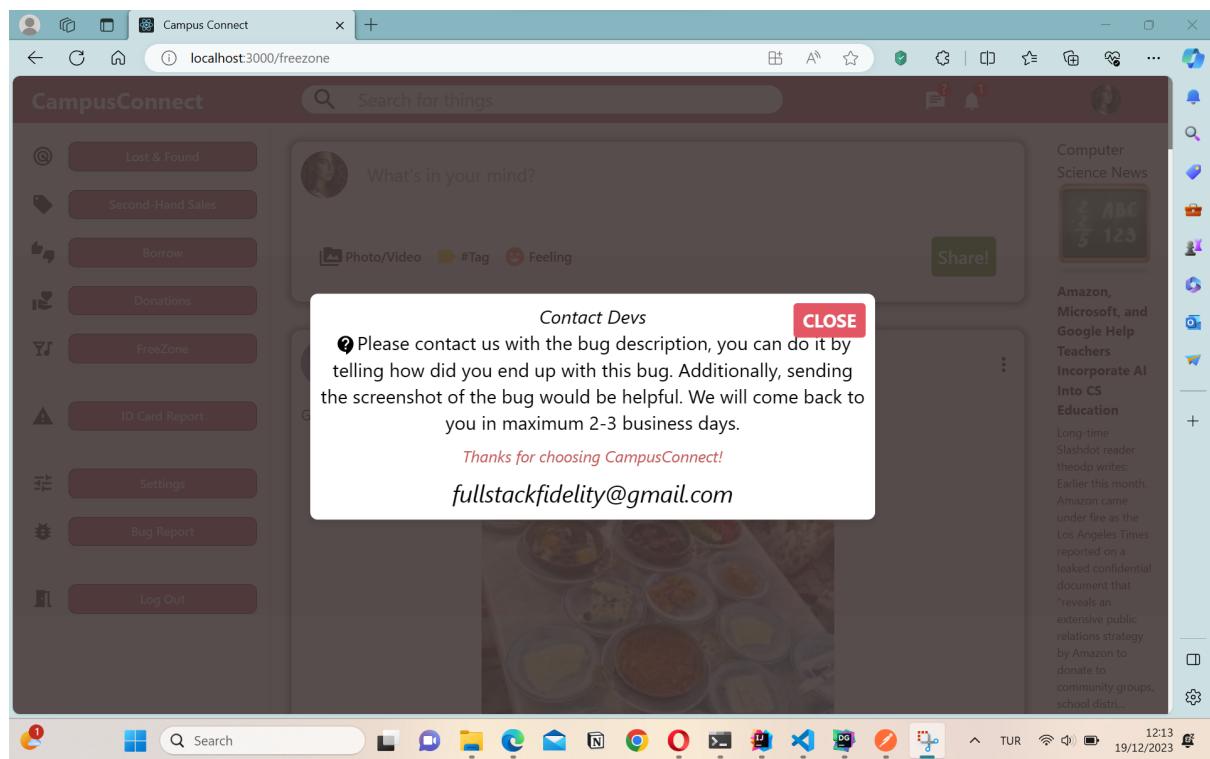
The user can also make comments on the Freezone posts.



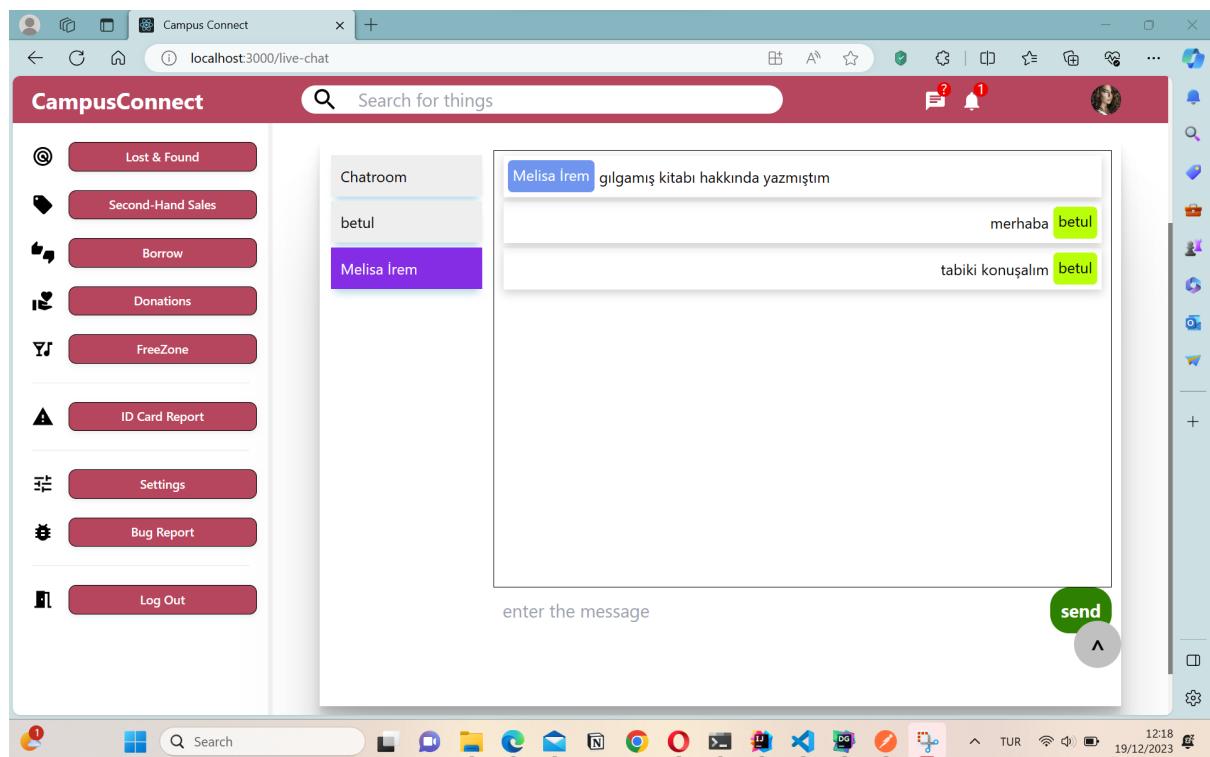
The following pop-up appears when the user clicks the repost ID button.

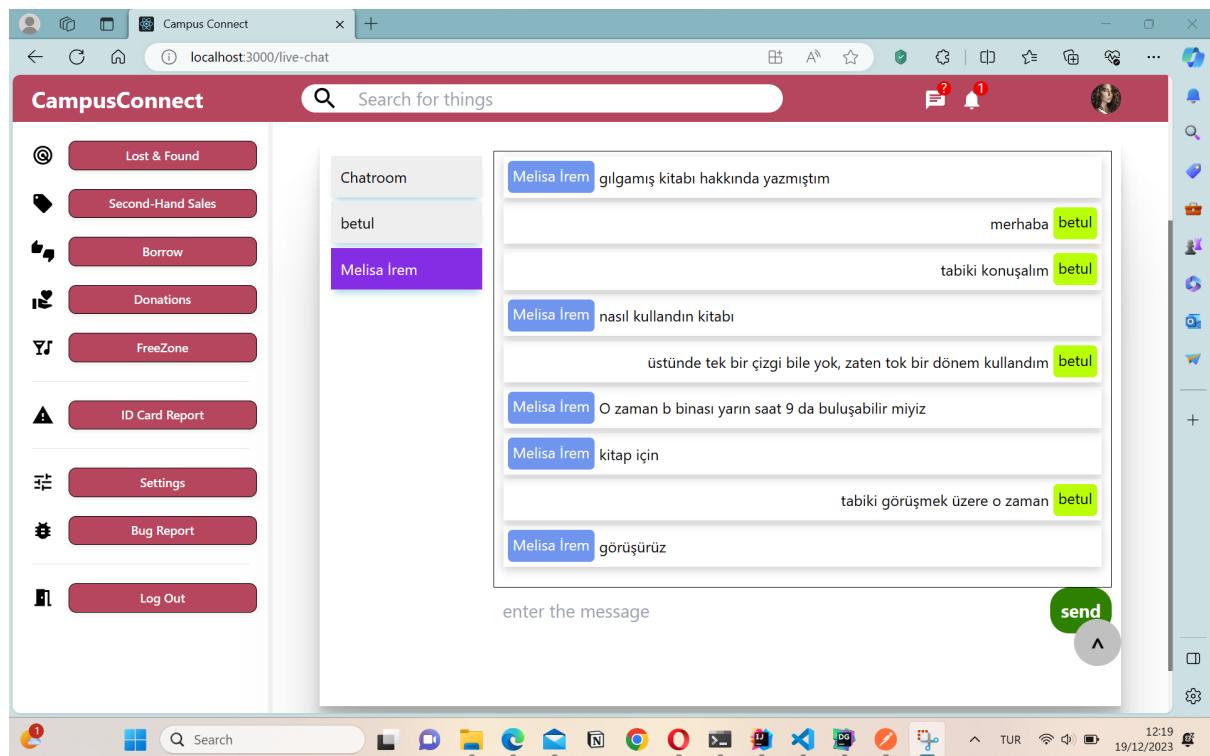
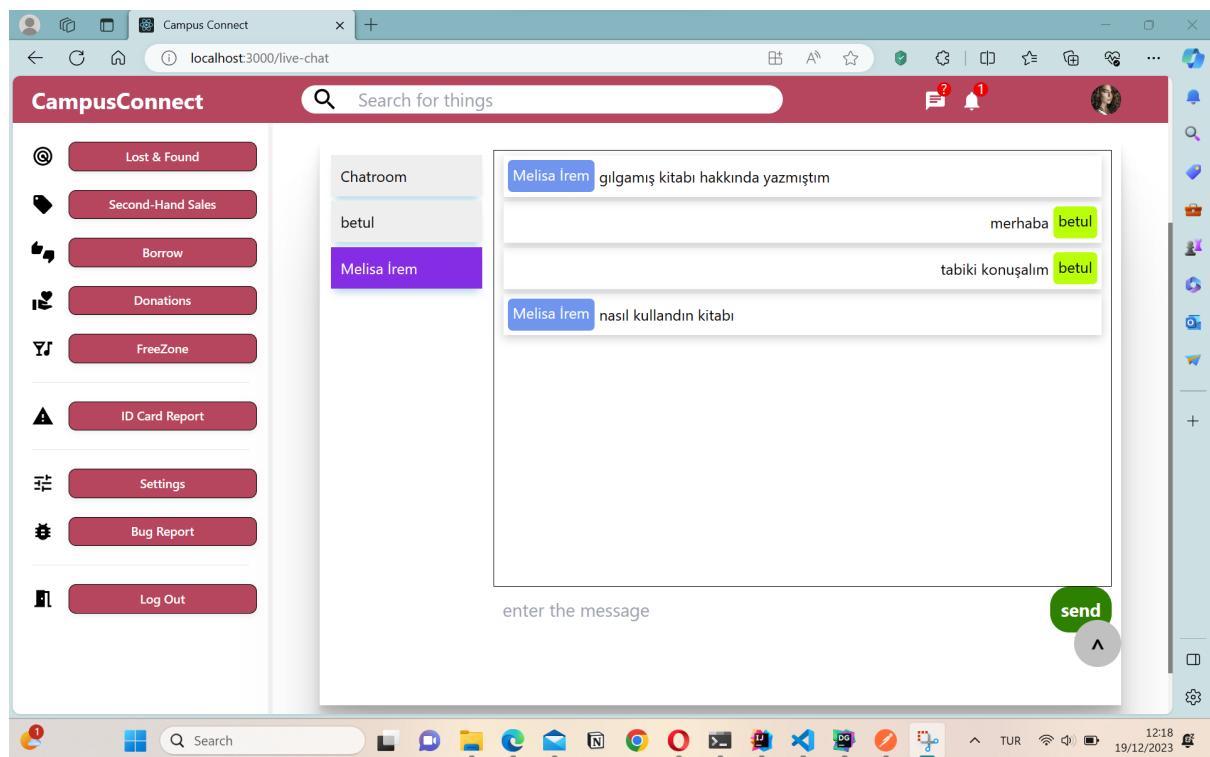


This pop-up appears when the user clicks the Bug Report part. The link here just directs to the mail for the user to compose.



The following are the live chat parts. Two users can contact each other in a live manner. The Chatroom is implemented for a public chat. All the users who are live at the moment will see the chat.





3.2 High-Level Description

CampusConnect is a social media platform that can solve university students' problems on campus and help them socialize. CampusConnect will serve as a savior platform for students when they find a lost object on campus, want to lend their course materials, sell their belongings second-hand, donate their excess materials, or socialize using FreeZone. It is a good choice with its easy interface for users to communicate more quickly with their university students when necessary.

3.3 How to Install the Software

WebSocket, Java, HTML, JavaScript, TypeScript, Cascading Style Sheet, Postman, Spring Boot (Spring Data, Spring Web, Spring Security), Maven, Lombok Library, PostgreSQL, React, Mui Library, Axios, Babel, IntelliJIdea, DataGrip, Visual Studio Code

The tools above could be accessible via their installation websites. Spring Boot projects could be initialized from its specialized website for creating projects. Libraries like Lombok can be added to the project as properties and dependencies.

3.4 How to Run the Software

After pulling the software from GitHub, the file under CampusConnect should be open in an environment like IntelliJ for the backend. For live chat in IntelliJ, the compiler recommends inside the “npm install” comment that it should be approved. If needed, Java version 17 should be specified to IntelliJ. After making the installments, the run button can be clicked. After that, for the frontend portion, after running the “npm install” and “npm build” commands, the required nodes will be installed in the IDE. Finally, as CampusConnect uses a cloud database as a user, it is unnecessary to handle the application's database part. For the frontend, to run the React user interface, the “npm start” command should be initiated to the terminal. After that, the application will be opened automatically, and the user can use it.

3.5 How to Report a Bug

A well-constructed bug report should be concise, comprehensive, and understandable. To write a good bug report, the following items should be included:

Title/Bug ID: It makes it easier to understand the problem.

Environment: The environment in which the project operates plays an essential role in solving the problem. When solving the problem, considering situations such as Device Type, OS, Tester, Software version, Connection Strength, and Rate of Reproduction helps solve the bug in specific ways.

Steps to reproduce a Bug: It is easy to follow specific steps to see the error and resolve the bug quickly and accurately.

Expected Result: This component of the Bug Report describes how the software is supposed to function in the given scenario. This helps them measure how much the bug disrupts the user experience.

Actual Result: Detail what the bug did and how it distorted the expected result. The specificity in this section will be very helpful to developers. When doing this, clearly highlight what went wrong and provide additional details so they can begin investigating the issue, considering all variables.

Visual Proof (screenshots, videos, text) of Bug: The visual evidence provided makes it easier to understand where and how the bug affects. In this way, the codes to be looked at become more specific.

Severity/Priority: Each bug should be assigned a severity level and corresponding priority. This reveals the extent to which the bug affects the system and, therefore, how quickly it needs to be fixed.

3.6 Known Bugs

3.6.1 Donation, Borrow, SHS Item Share is not working

This has been implemented in the backend but could not be connected to the frontend due to time constraints.

3.6.2 Comment liking is not working

The likes for the comments cannot be saved into the database, although they are shown in the frontend.

3.6.3 Freezone Share User Profile Picture is buggy

When a post is shared in Freezone, the shared post gets the last post sharer's user in frontend. The issue can be solved by refreshing the page once.

4. Developer-Manual

A detailed guide for developers contributing to the project. This manual will cover technical aspects, coding conventions, and guidelines for maintaining and extending the project.

4.1 How to Obtain the Source Code

To obtain the source code, clone the repository at the following link:
<https://github.com/CS319-23-FA/S3T3-FullStackFidelity>. You must be in the 'final' branch of your local machine.

4.2 The Layout of Our Directory Structure

The layout of our directory structure is as follows:

At the top, we have a directory containing the front and backend directories. The name of our backend directory is CampusConnect, and the frontend directory of our app is log-reg. Inside the CampusConnect directory, all our source codes are inside the 'src' folder. In the src folder, we have a main directory and a directory where the backend testing occurs. Inside the main directory, we have two directories. One is 'resources' that contains the 'applications.properties' file, and the other contains our backend codes. Inside the fullstackfidelity folder, we have DTO, chat, controller, entities, exception, interfaces, repositories, service, and util directories. The chat directory is formed to properly view the WebSockets, their configuration, and their use. The chat directory contains a configuration, a controller, and a model directory. The config directory only includes the WebSocket configuration class, the controller directory comprises a controller for the chat WebSockets, and the model is where the message entities are held. The rest of the packages in the fullstackfidelity folder are designed to have an ordered packaging to have a model-view-controller architecture. The 'controller' directory contains all the controller classes of the usercontroller, itemcontroller, postcontroller, etc. The 'entities' directory has the entity classes held in the database, such as item class, user, and class. There is the repositories directory, which controls the connection of the entity classes with the database, and there is a directory called service, which controls the transactional operations and business logic. It acts as a connection between the user interface and the data access layer. The interfaces directory contains the relevant interfaces for our classes, and the exception directory contains the exception classes to include the edge cases in our application. The last directory is the util directory, which compresses the images uploaded from the application. All of our frontend code is in the 'logreg-page' page. The public directory inside is for icons and pictures uploaded. The 'src' directory ID is for all the frontend components and their interactions. The components directory includes dashboard, assets, forgotpasswordpage, general and loginsignup directories. The assets directory has images uploaded for the items. The dashboard directory contains code for the mainpage. The forgotpasswordpage contains code for the forgetpasswordpage frontend codes. The loginsignuppage directory contains JSX files for the login and sign-up page. The general directory is our main directory for the frontend. It includes pages and component directories. The components directory includes general modal components. The pages directory manages the flow between different pages and contains the building blocks for the pages.

4.3 How to Build the Software

4.3.1. Backend Instructions

To obtain the source code, clone the repository at the following link:
<https://github.com/CS319-23-FA/S3T3-FullStackFidelity>. You must be in the ‘final’ branch of your local machine. Make sure you are on the “main” branch. The backend code of our application is in the ‘CampusConnect’ directory, and the frontend is in the ‘logreg-page’ directory. The backend should be from an IDE such as IntelliJ Idea. To open the project, open IntelliJ Idea Ultimate (this is preferred). The first thing you need to do is to set up the SDK for the project. The SDK of the project should be “OpenJDK-20 java version 21.0.1”. Check the project SDK from File -> Project Structure -> SDK. If it is something else, you can download it from Project Structure -> Platform Settings -> SDKs -> Add new (“+” button) -> Download JDK. If there's no build configuration, select Run -> Edit Configurations -> Add new (“+” button) -> Spring Boot and Maven. Also, you need to accept the download when IntelliJ asks you whether to do an npm install. After the configurations are done, the backend should be built and runnable.

Our project database is on a ‘Render’ server online, a PostgreSQL database. Therefore, you must do nothing to connect the database to the application. It is already connected to the application.properties file. You can see the database from the DataGrip application of IntelliJ by selecting to add a database and using the following credentials to know the database:

hostname:dpq-clpi781oh6hc73c43uag-a.frankfurt-postgres.render.com

port:5432

username:database_520z_user

password:QJZRrkHF8ZkLixWk3e3fZYBVBSdHZoB

After that, when you open the folder com.fullstackfidelity.campusconnect.CampusConnectApplication and push the run button, the server is instantiated and can be used.

4.3.2. Frontend Instructions

To run the frontend, you must install some dependencies. After you open the log-reg folder in an IDE, write the following commands in the terminal.

“npm install,” “npm install stompjs,” “npm install sockjs-client,” and “npm install net.”

After installing these dependency packages, you need to run the application to start the frontend part of the application. You can run the application by writing “npm start” to the terminal.

5. Work Allocations

All participants worked on writing the reports and drawing diagrams.

Süleyman Yağız Başaran 22103782

I was responsible for the frontend of the program, nearly the whole frontend. Details will be at the bottom. I used React and classic stuff CSS with HTML. The application starts from the login, register, and forgot password pages. I was also implementing axios for these pages to make it easier to connect with the backend. After logging in, the main page with delete the account and go to Freezone buttons are made. I made the welcome page inside the site with our logo, left-bar, right-bar, top-bar, and every component inside it. I also linked these pages with react routers. I wanted to get a vision similar to that of earlier versions of Facebook. After the components in all the pages, as I mentioned. I first created the Freezone feed. Inside it I created a share box with all its buttons: add photo/video, tags, reaction emojis, and share button. After that, I created post components in the freezone. It has a name, profile picture, post date, picture, three dots at the right corner, comments, and like function. You can like or dislike it in the frontend and it shows. I created different 4 feeds for the sections lost and found, second hand sales, borrow and donations. I implemented masonry grid for them. Also in the feeds we have + button which creates posts for the selected feed and the title. Then I used MUI Cards inside these feeds. I changed them to work with pop-up(modal) and added hovers. Lost and found has extra input which is location and it shows. Second hand sales has price and age of the item. Borrow items has given date and desired taken date. Every single of them has name, description and modals. Lastly, I created bug report modal. Briefly, most of the front end was done by me and also functions were done by Serhat which helped me a lot.

Gülbera Tekin 22003354

My main contribution to this project is the backend of Items. I created and edited entities, repositories, services, and controllers for each item type and IdCardLost feature. These items could be creatable, deletable, updatable, for BorrowItems borrowable, returnable, holdable, for DonationItems donatable, for SecondHandSaleItems sellable, buyable, for LostAndFoundItems reportable as found, and for IdCard reportable as IdCard found. Besides these features, I made tests for the backend and database connection to ensure that item classes work correctly. In addition, I added code for FileSystem to store images to speed the database up. I also helped my friends with other backend sections like LiveChat and UI materials, such as creating the logo.

Onur Tanınmış 22003312

My contributions to the project were both in the frontend and the backend. I helped Gülera create and develop the features of the classes related to donation items in the backend and helped design the logos and badges in the frontend. I also helped my friends in connecting the frontend with the backend.

Serhat Yılmaz 22002537

I contributed to the project's frontend part with Yağız. I added share post and post new items and worked on the profile page. I handled onClicks (functionalities) of the buttons and texts. The change profile picture button disappears when clicking on other people's profiles. When you click on the chat buttons, you can go to LiveChat. Moreover, I did modal designs. I helped my friends connect the backend and frontend with Axios in React. Lastly, I handled the routes of the pages in the frontend part.

Melisa İrem Akel 22003923

With SpringBoot, the following classes, and their services and controllers are created: Notification, Post (this is for the FreeZone part of the application), Comment (the comments of the FreeZone posts), and User. I can say that I did most of the connection of the backend with the frontend using Axios.

Details:

The FreeZone posts will be filtered if a user blocks another user. Moreover, the posts are fetched with pageable, which allows faster fetching with many posts, delete and update the FreeZone post if the user is the same as the post's user, and FreeZone uses infinite scrolls that fetch data by scrolling and reserving the y coordinate of the user is also implemented by me. Moreover, I added additional computer science news by using a ready API.

Notifications will be sent if a user loses their Bilkent IDs to protect the privacy of the users. I implemented user notification in backend coding and its connection with the frontend.

Comments classes are mainly for Posts. Moreover, users can reply to each other in the comment section.

The connections of item classes and Freezone to the frontend with the backend.

Searching functionality for items and FreeZone posts.

Adding skeletons for items and FreeZone parts to have a better visual.

I implemented the search bar part of the application that filters items with their name and FreeZone posts by their contents.

Betül Doğrul 22003559

My main contribution to the project is the live-chat part. I have implemented both the backend and the frontend. The controller, configuration, and model classes. I have used websockets and sockjs for the live chat. The chatroom feature of the live chat is implemented so that all the online users can interact with each other. Also, the users can send private chats to specific users. Also, I have moved the database to a server through the render application, making it easy for us to debug and enhance the application. I also contributed to the database connection and debugging of the database and backend. We (backend developers) used Postman and Datagrip to see the database connections for debugging purposes. I also helped my friends by connecting the frontend with the backend and debugging the entity classes.