CS 319 Object-Oriented Software Engineering

Deliverable 6

Final Submission

S2T8

Bilginer Oral	Mohamed Amine Boukattaya	Eren Karakaş
22103163	22301131	22002722

User Documentation

High Level Description

CampusConnect is designed to be a one-stop solution for students' various academic and social needs. We have observed that previously existing platforms, such as Instagram (e.g., Bilkent_itiraf_ediyor) and Facebook (e.g., Bilkent Duyuru), were insufficient in meeting the diverse needs of Bilkent members. Members of our community previously found themselves navigating a cumbersome process, having to post items they had lost on Instagram, then switching to Facebook to create another post for their various needs, then switching to LetGo in search of affordable second-hand items, etc. In response to these challenges, CampusConnect takes these ideas and not only refines them but also enhances their functionality by integrating them into a unified platform. By centralizing these crucial functions into a single platform, CampusConnect seeks to enhance the lives of members of the Bilkent community, offering them an integrated solution that seamlessly caters to their academic and social needs.

CampusConnect isn't just another app; it's a transformative experience for Bilkent University students. It empowers them to build meaningful academic and social connections in a secure, user-centric environment. CampusConnect is designed exclusively for the Bilkent University community, which adds a sense of exclusivity and security. Users can connect with their peers, faculty, and staff in a closed network, fostering a sense of belonging and trust. One of the most appealing aspects of CampusConnect is its ability to integrate a wide range of features into a single platform. It caters to various aspects of university life, including lost and found items, second-hand sales, borrowing, donations, and real-time messaging. This integration makes it a one-stop solution for the diverse needs of Bilkent University students. We believe that with this app, students can seamlessly blend their academic and social spheres, ensuring that their university experience is not just productive but also memorable and fun. CampusConnect isn't just about connecting virtually; it's about creating a vibrant and supportive campus community both online and offline, all in one place. The app, moreover, supports the following features:

🔍 Lost & Found Items

The Lost & Found Items feature allows users to report lost items and post found items within the school community. Users can post details about their lost belongings, such as location, description, etc. while also being able to post information about items they have found. This feature helps the community reunite with their lost possessions. CampusConnect aims to replace the old-school Instagram pages by providing a user-friendly and secure environment.

💰 Second-hand Sales

The Second-hand sales feature enables users to list items they no longer need or want to sell within the school network. This feature encourages sustainable and budget-friendly exchanges of items like textbooks, electronics, furniture, and more among students and staff. The fact that CampusConnect can only be used within Bilkent makes CampusConnect preferable over other platforms such as LetGo in terms of providing a secure environment.

🤝 Borrowing

The Borrowing feature allows users to lend and borrow various items or resources from fellow school members. The range of the items borrowed can vary from study materials to sports equipment.

🎁 Donations

Using the Donations feature of CampusConnect, users can donate items, funds and possibly anything to individuals within the school community. Whether it's contributing to a charity drive, supporting a student in need, or donating used clothing, this feature promotes a culture of giving and support within the school.

💬 Real-time Messaging

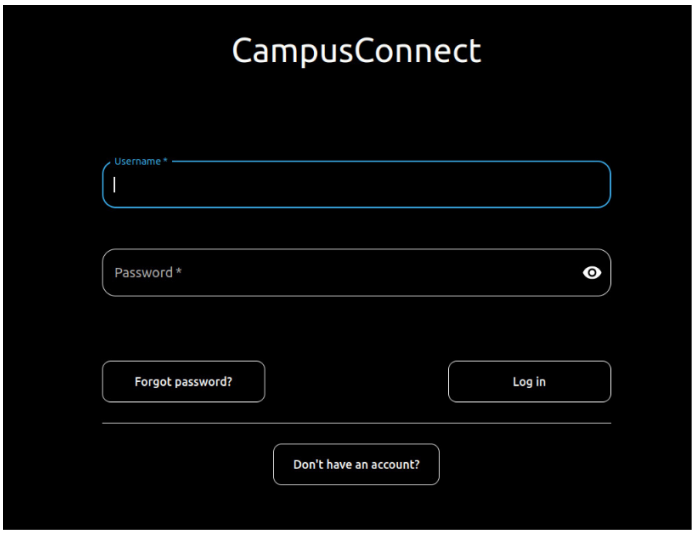
The Real-time messaging feature enables instant communication among school members; allowing students and staff to discuss assignments, coordinate group projects, or simply stay connected with peers in real-time.

How to install the software.

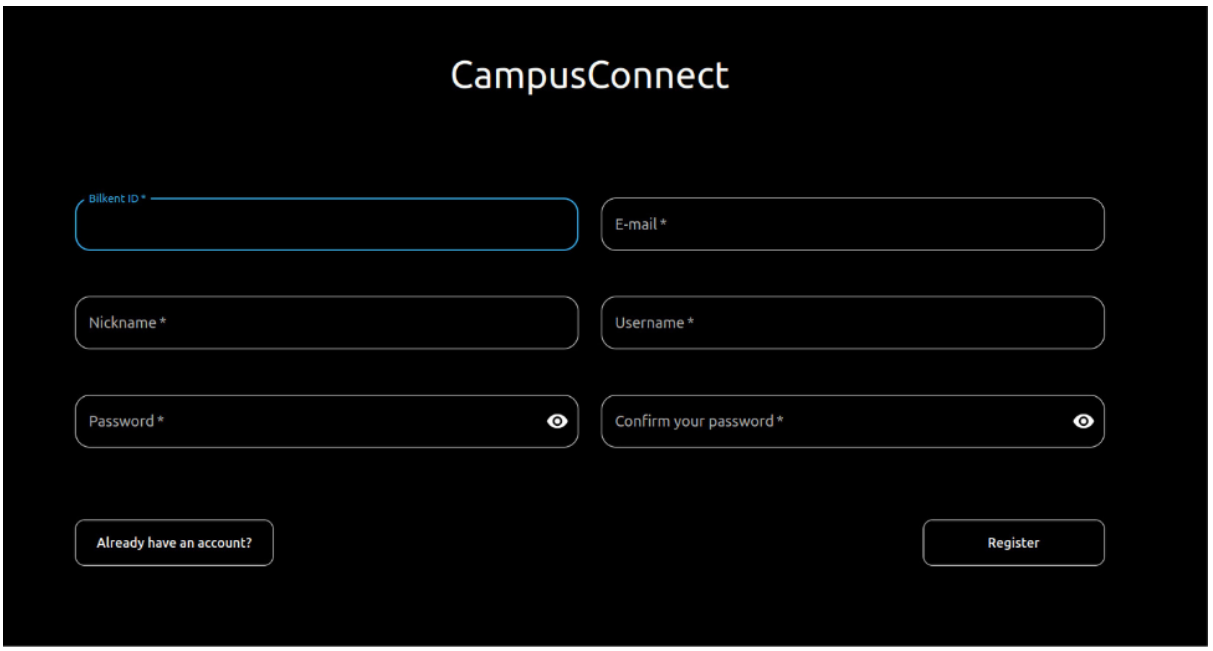
Since CampusConnect is designed to be a web application, no prior installation is required for its users to use the app, except a web browser. Because the application isn't hosted online, this approach is a bit theoretical however. To build and run the application on a localhost, follow the instructions on the Developer Documentation.

Using the Software

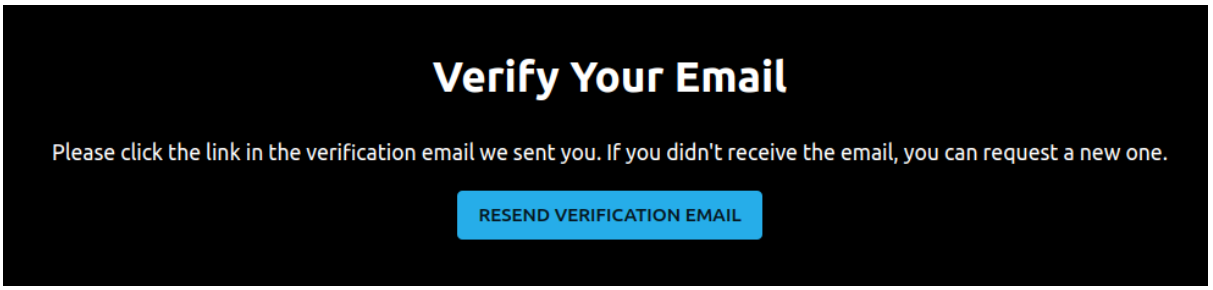
Authentication



First, create an account by clicking "Don't have an account?" button. If you already have an account, enter your credentials and click the login button.

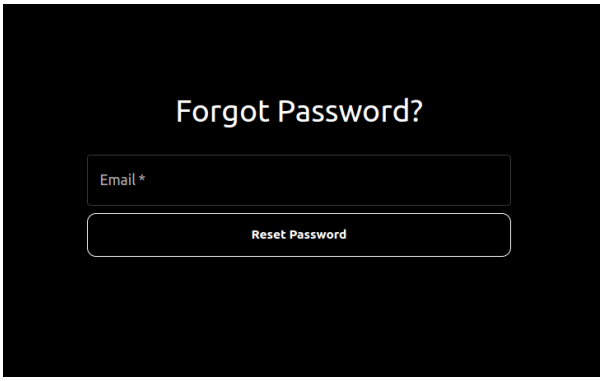


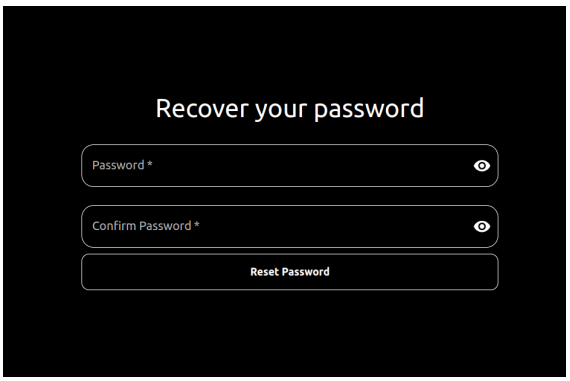
After clicking on the “Don’t have an account?” button, you will be redirected here where you have to type your Bilkent ID, your email, nickname and name in the indicated spaces. After carefully choosing your password and confirming it, you click on register.



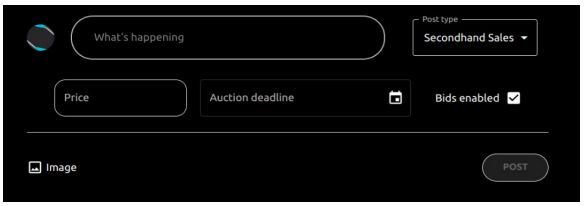
You have to go to your mails and click the verification link afterwards. Otherwise it will not let you in. Once you verify your email, log in with your credentials.

If you happen to forget your password, you can click the “Forgot Password?” button. You can then enter your email you used on registration, and click “reset password.” This will send a unique link to your mail account. Clickin that link, you will be directed to the password recovery page, in which you can update your password.

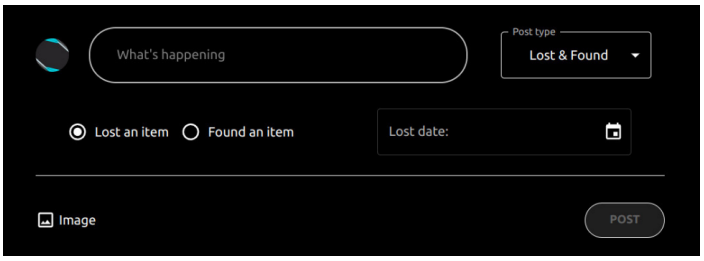




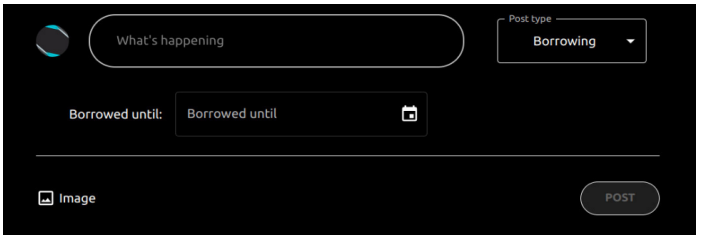
Uploading Posts



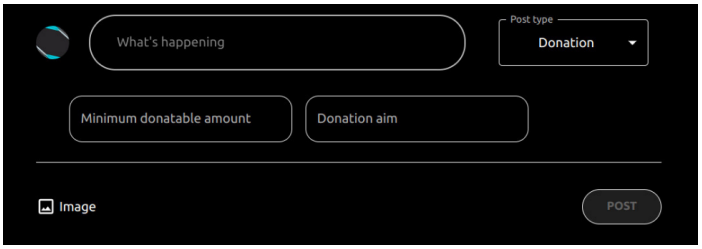
To create a secondhand sales post, first navigate to the website's homepage. In post type select “Secondhand Sales“. Here, you'll start creating your sales post. You will need to provide details about the item you're selling, including a description and images. When it comes to pricing, you have two choices. If you want to set a fixed price, simply enter the amount in the designated field. Alternatively, if you prefer to auction the item, select the "Bids enabled" option. For auctions, you must set a starting price and specify the end date for the bid. Once all details are entered, click the "post" button to post your sale.



To create a “lost and found” post, first navigate to the website's homepage. In post type select “Lost & found“. Here, you'll start creating your lost & found post. You'll have two main options to choose from: either you have lost something or found something. Select the appropriate option. Then, provide detailed information about the item, including its description, and the date and location (in the text field) of when and where it was lost or found. Adding images of the lost or found item can significantly increase the chances of a successful match. After filling in all the details, finalize your post by clicking on the "post" button.



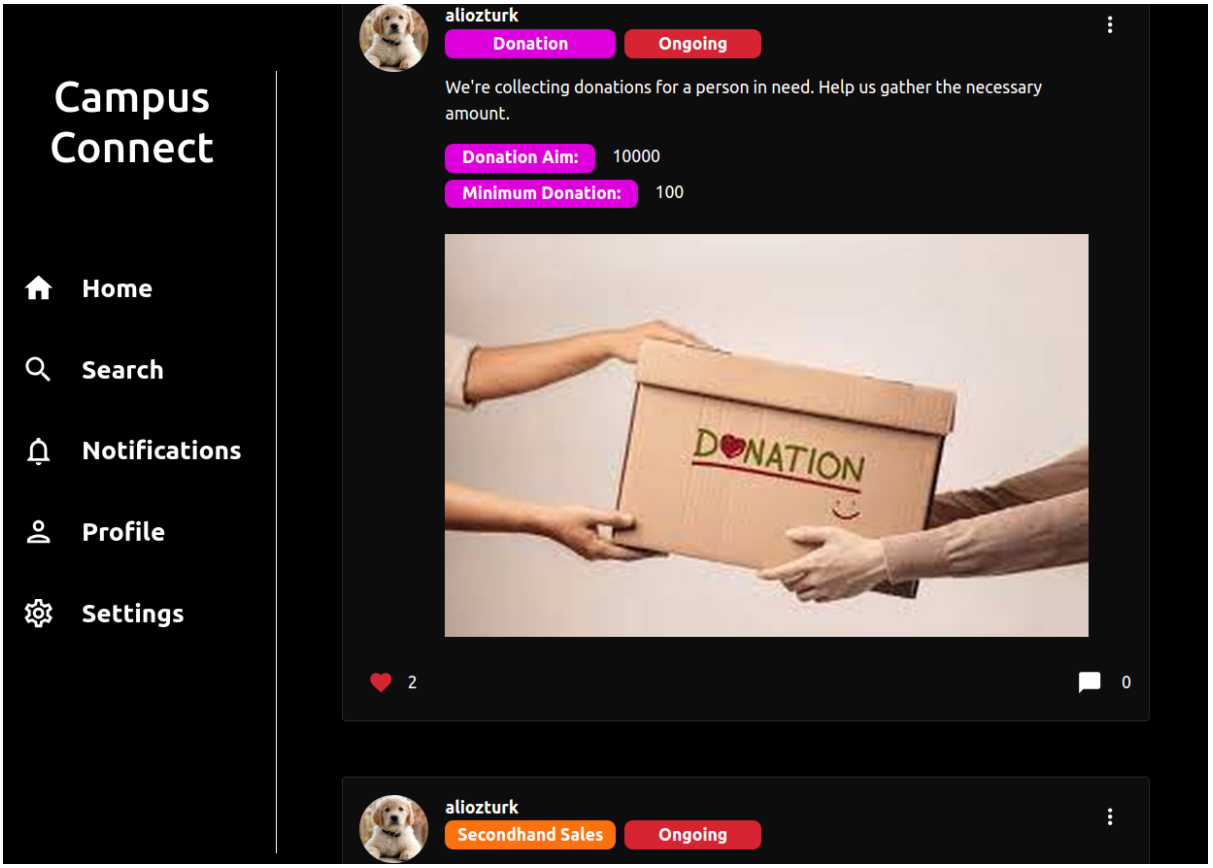
To create a “Borrowing” post, first navigate to the website's homepage. In post type select “Borrowing“. Here, you'll start creating your lost & Borrowing post. In the post creation interface, you will be required to detail what item you wish to borrow. Additionally, it's important to state clearly when you intend to return the borrowed item. This information helps lenders understand the terms of the borrowing. You can also add photos of the item if you have specific requirements or preferences. Once you've filled in all necessary information, submit your post by clicking the "post" button.



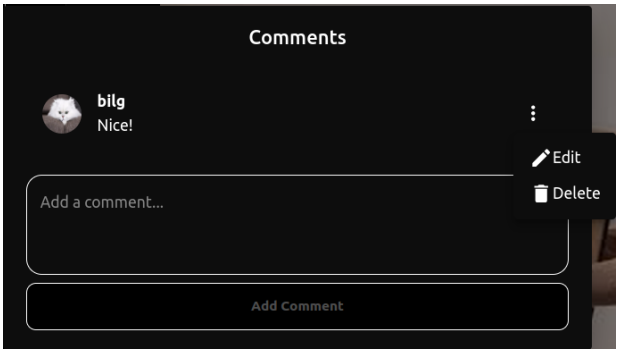
To initiate a donation post, first, go to the main page of the website. in Post type choose “Donation”. In the posting interface, you'll be prompted to specify the "Minimum donatable amount," ensuring donors know the least amount they can contribute.

Additionally, in the "Donation aim" field, describe the purpose of the fundraiser and how the funds will be used. If you wish, you can add visual context by clicking on the "Image" icon to upload a relevant image. With all information correctly filled in, finalize your donation post by clicking on the "Post" button.

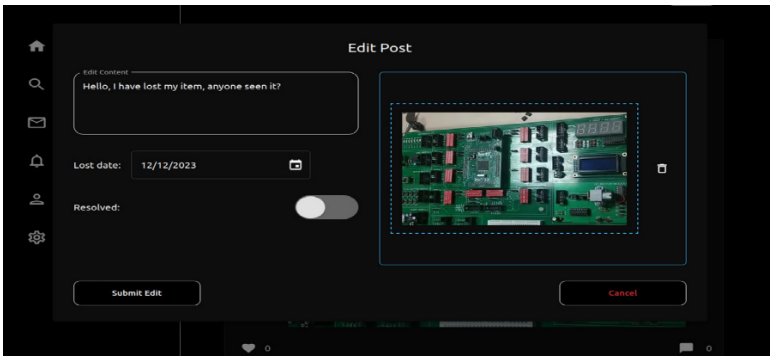
Homepage



In the homepage, you can see your own and others' posts. You can like them and comment on them. And you can edit and delete your comment too.

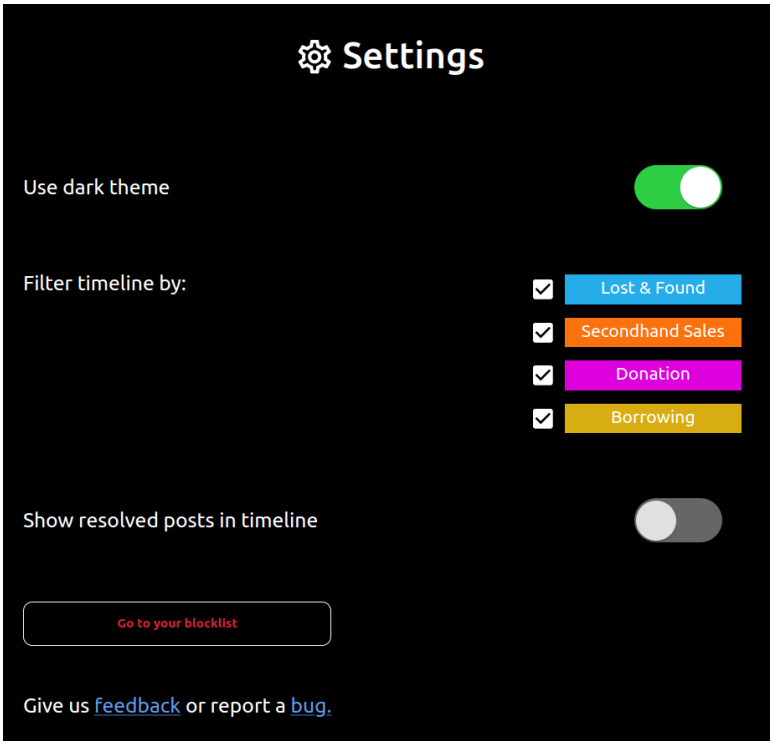


You may also edit your post too, by similarly clicking on Edit button.



Here we chose a lost & found post as an example, but you can see that you can modify anything or mark it as resolved; which means the item was found in this case.

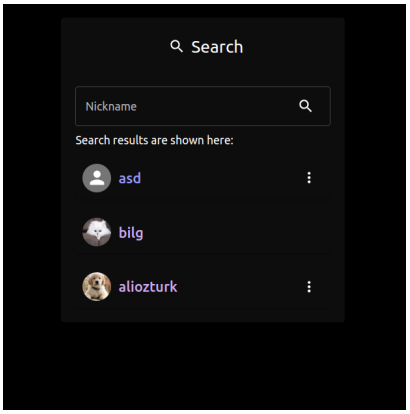
Settings Page



This is the settings page of the user, in which:

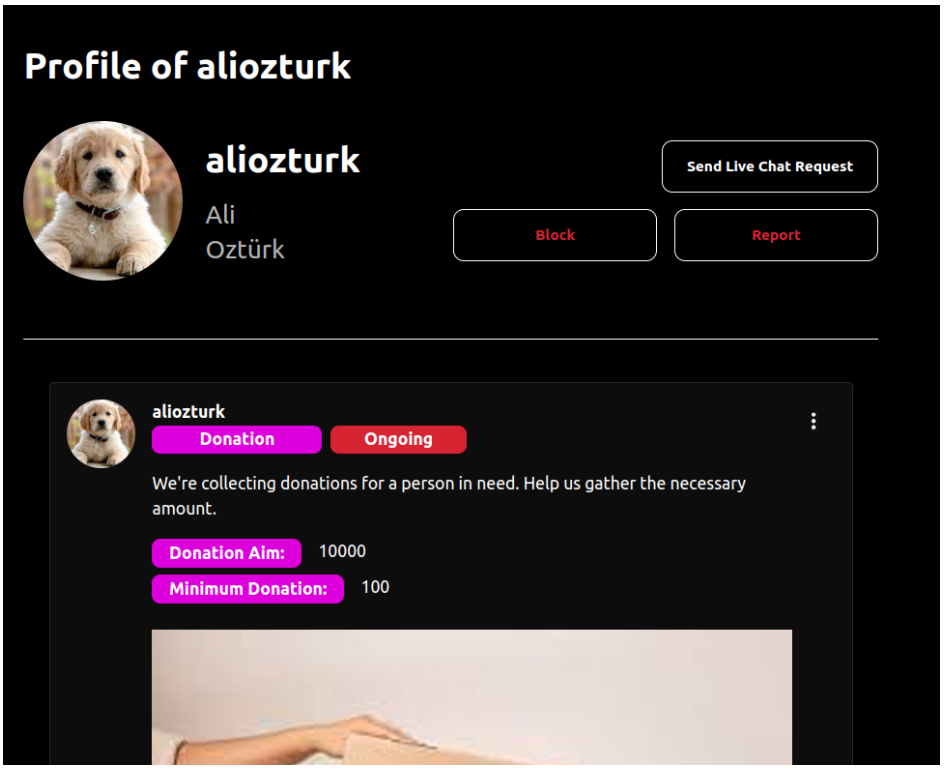
- You can change the theme of the website
- Filter the timeline by what suits you
- Select whether or not to show resolved posts
- You can acess your blocklist
- Give feedback or report a bug

Search Page

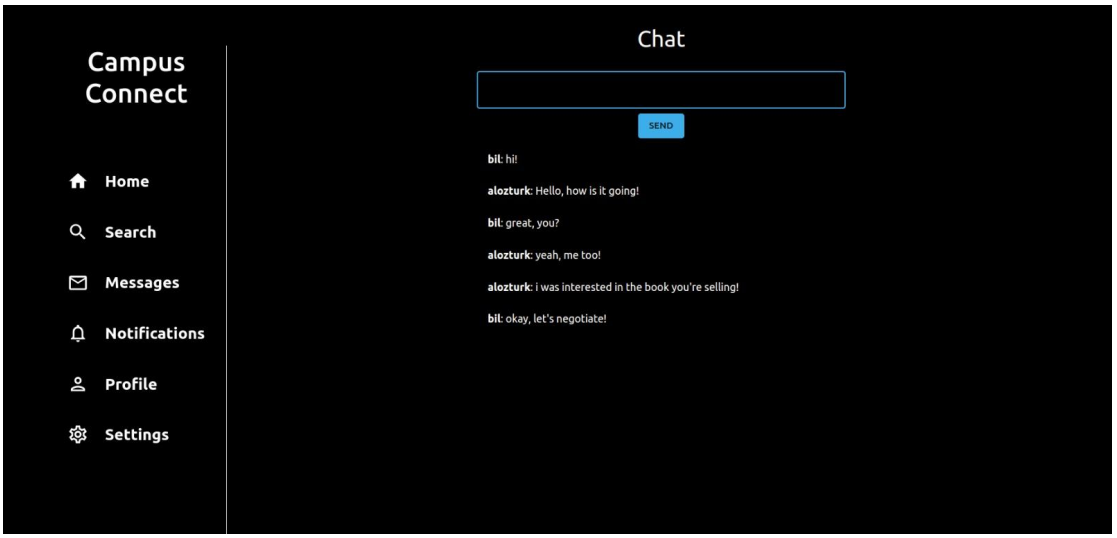


On the search page, you can search for users and go to their profile.

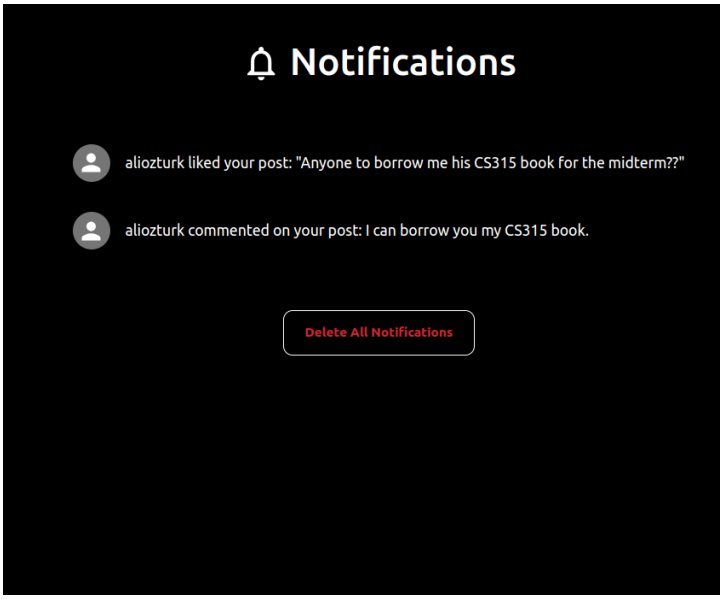
User Profile



You can also search through other users' profiles. Similarly you can like their posts and comment on them. You can also block, report, and send live chat requests to them. If accepted, the live chat request will initiate a live chat.

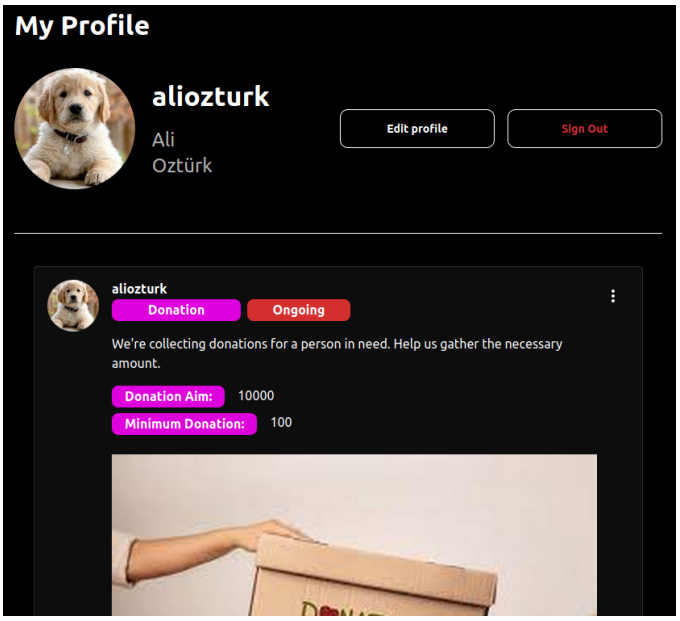


Notifications Page



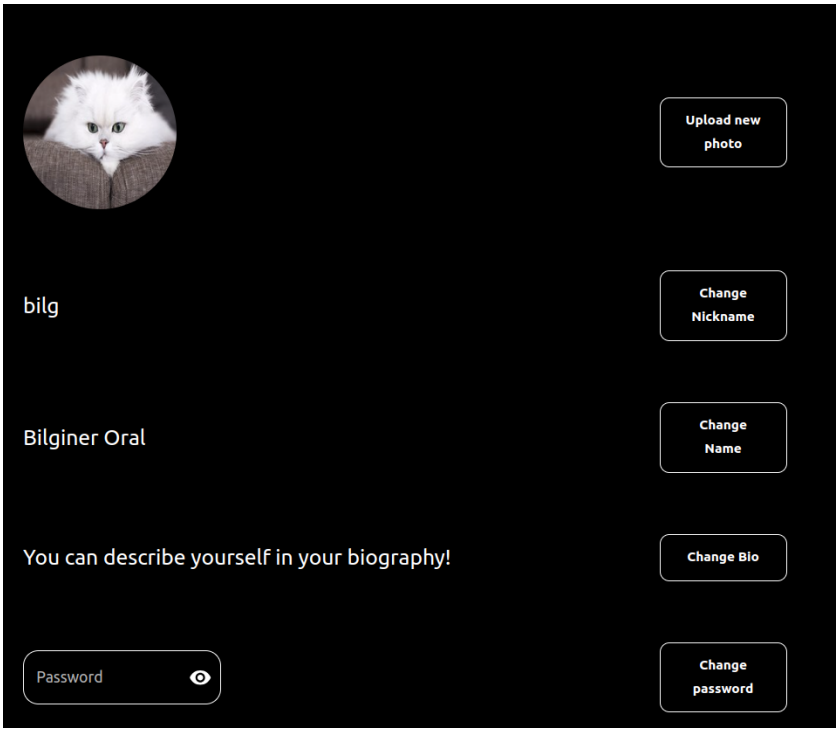
In the notifications page, you can see your notifications which are created when a user likes your post or comments on it.

Own Profile Page



You can see your own posts in your profile. Also, clicking “edit profile” will direct you a new page.

Edit Profile Page



You can view your profile settings an update them as you wish in the “edit profile” page.

Issues and How to Report Them

In the Settings page of our application, you can click on the “report a bug” link, which will take you to a Google Form with preset questions. You must fill in the following fields:

1. Title of the bug report
2. Environment being used. This information includes, but is not limited to, the web browser used, hardware specifications, and the your operating system
3. Steps the reproduce the unintentional behavior. Here describe how the developers can produce the same behavior to track what causes the issue.
4. Expected vs. actual results
5. Additional information

Known Bugs

Known bugs of CampusConnect currently includes:

1. Random WebSockets disconnects.
2. UI changes position slightly when clicked on particular buttons.
3. Making bid rarely fails.
4. Editing own profile sometimes fails.

Developer Documentation

The project can be found at the following link:
https://github.com/CS319-23-FA/S2T8-bilcoders

You can copy the source code of our project to your local machine using the following command:

git clone git@github.com:CS319-23-FA/S2T8-bilcoders.git

(If you don't have an SSH key, you must create one first. Follow the guide here for step by step instructions: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)

1. Layout of the Project

From the root of the project, the app folder contains the source files that detail the back-end code, the frontend folder contains the source files that detail the front-end code, the images folder contains an example documentation image, and the Documentations folder contains the project deliverables done until now.

In the app folder, the database folder contains the source files implementing the database models (users, posts, etc) and the DatabaseManager class. Similarly, the chat folder contains source files implementing the WebSocket manager and ChatManager classes. Other files in the app folder remain as stand-alone files (e.g. authentication.py file contains the AuthManagar and email.py file contains the EmailManager).

In the frontend folder, the src folder includes all the React.js source files. Inside, the components folder includes the source files that implement components; reusable and independent UI pieces. The pages folder includes the source files that implement the app's pages. Pages are contained inside folders, sometimes by themselves and sometimes by other auxiliary pages.

2. Building the Project

Firstly, you need to clone the project repository as documented previously.

The app folder contains our back-end code, and the frontend folder contains our front-end code.

For the front-end:

1. Navigate inside the frontend directory using your terminal emulator.
2. After making sure you have the npm application installed, run "npm install" to install the dependencies:
3. Now, you can run the front-end portion using "npm run dev".

For the back-end:

1. Navigate inside the app directory using your terminal emulator.

2. We recommend creating a Python virtual environment for a better development experience (using either venv or conda). Python 3.9+ is required.
3. After you activate the virtual environment, run the following command to install the dependencies:

“pip install -r requirements.txt”. Remove the line "uvloop==0.19.0" from requirements.txt if your development environment is currently on Windows.

1. Then, inside the app directory, create a folder named "static". Inside static, create two more folders named "profile_images" and "post_images".
2. In order to use the database and more functionality, you need to create a ".env" file in the project root directory. You must populate the .env file in the following way:

EMAIL = “”
PASSWORD = “”
DB_URL = ""
SECRET_KEY = ""
ALGORITHM = "HS256"
ACCESS_TOKEN_EXPIRE_MINUTES = 30

1. For the EMAIL field, you must enter the address that EmailManager will use to handle email operations.
2. For the PASSWORD field, you must provide the application password for the given email address. Follow this guide on how to create application passwords for a gmail account: https://support.google.com/mail/answer/185833. This functionality has not been tested with other mail providers.
3. For the DB_URL field, you must enter your PostgreSQL database URL, which also contains your database username and password. The format for the PostgreSQL URL is "postgresql://username:password@hostname:port/db-name". To fill the SECRET_KEY field, run "openssl rand -hex 32" in your terminal application and paste the output in between the quotation marks.

For detailed instructions on how to setup PostgreSQL for running the app, see https://www.guru99.com/download-install-postgresql.html.

Now, you can run the back-end by using the following command while you are in the app directory:
uvicorn main:app --workers 1 --host 127.0.0.1 --port 8000

1. Once the server is started, you can go to "http://localhost:8000/docs" address to see the API endpoints and their documentation. You can also test the API calls here.