

# **CMPE 451**

## **Milestone 2 Report**

---

### **Group 9 - Postory**

---

Mert Alkan

Ahmet Melih Aydoğdu

Zehranaz Canfes

Mustafa Emir Çolak

Ömer Barış Erkek

Melih Özcan

Ahmet İbrahim Şentürk

Niyazi Ülke

Tag

# Table of Contents

---

## Table of Contents

### Executive Summary

- Introduction
- Project Status and Work Done So Far
- Moving Forward

### Implementation of Core Functionality

- Frontend
- Android
- Backend

### Satisfaction of Requirements

- Summary
- List of addressed requirements
  - Requirements in Progress
  - Explanations
  - Completed Requirements
- List of unaddressed requirements
  - Unaddressed Functional Requirements
  - Unaddressed Non-Functional Requirements

### API Endpoints

- Sign Up
- Activation
- Sign In
- Post Create
- Post Update
- Main Page Posts
- Discover Page Posts
- Specific User's Posts
- Profile photo
- Comment
- Like
- Follow User
- View a Single Post
- View a Single User

### User Experience Pages For The Web

- Screen shots for create/edit post pages
  - Link to code
  - Adding Story
  - Adding Time
  - Adding Locations
  - Adding Tags
  - Seeing the preview
- Sign-In page
  - Link to code
- Sign-Up page
  - Link to code
- 'Forgot password' page
  - Link to code
- Profile page
  - Link to code
- Discovery page
  - Link to code
- Viewing posts page
  - Link to code
- Main page
  - Link to code

## User Experience Pages For The Mobile Application

Login Page

Link to code

Signup Page

Link to code

Main Page

Link to code

Profile Page

Link to code

Pick Image

Link to code

Edit Posts

Link to code

Explore Page

Link to code

Create Post

Link to code

Single Post Page

Link to code

## Work Done by Each Team Member

# **Executive Summary**

---

## **Introduction**

---

As a group in CmpE 451 course, we aim to design a platform for people who want to share their stories based on locations they have been to and experienced. We are to implement a web and an Android application for our users to share stories with others and view them.

This milestone includes many new features. Main ones are users, locations and dates! Last two are the key functionalities of our app, which defines our project. The status of them is explained below.

## **Project Status and Work Done So Far**

---

In this milestone, the project plan was followed as we did in the previous milestone. In the first milestone, the connection between backend and fronted/android was established and the home page for our application was created. Posts could be created and edited.

User functionality is added in this milestone. In the first milestone, there were no users. Only the concept of home page and post creation were added. Now, users can sign up to create account and sign in the application. They can follow each other and see all public posts on the discovery page. They also see the posts of the users that they follow on their home page. Users can also comment and like other users posts and visit other users' profile pages.

Moreover, the location field in the posts is updated. Now, users can choose locations by clicking on a map. If they want to choose multiple locations, they can, and those locations are added in order so that the story can be more meaningful. The time field in the posts is also updated. Users can now add the time flexibly. They do not have to specify a certain time.

While implementing those features, frontend and android teams added new pages and backend team added new API functionalities and authentication systems. Those new pages are login page, sign up page, discovery page, new post create page, detailed post view page and profile page. For all those pages, new API functionalities, login and signup mechanisms are implemented by backend team.

After getting constructive feedback after the milestone 1 presentation, we reflect this feedback into our plan. We took the customer's feedback into consideration and implemented the features that the customer wants to see.

In conclusion, our application has now users, and those users can create posts flexibly. They can interact with other users by visiting their profile pages and following them. They can discover new posts via discovery page and they can also interact with posts by liking and commenting.

## **Moving Forward**

---

On our road to the second Milestone, we continued with the implementations. As we did our planning during the first milestone, we paid attention to the project plan. Each team tried to follow the plan which helped us prepare for the final product we will present during the customer presentation. We realized the importance of planning and following a plan. We will also continue our planning for the final milestone which we think is the most critical milestone.

Moreover, it is important for the android, frontend and backend teams to stay in contact and harmony. Since we worked towards the same product, each team should have the same goals. Doing regular team meetings, the tuesday labs, and many more communication via WhatsApp, we managed to take parallel steps towards our goal. We will also pay attention to our communication towards the final milestone.

After this milestone, we will implement saving posts, settings, report page. Most importantly, we will implement searching and filtering, which we think is the most challenging and crucial one. Our app relies on locations and their stories. So, it is important to create an interface for users to search and find the stories they are looking for. Users will also be able to see stories of other users as a memory book. With this book, users will be able to view each post in a locations with a flow. This way, the relation of time and location will be visible through our app, which we find fascinating.

Lastly, it is a known fact that implementing a project and maintaining it is a challenge. We will face several bugs and problems throughout this semester and the most important part while facing the problems is to stay as a group and work together. As we did during the first milestone, we will work hard and try to learn and improve as much as possible.

# Implementation of Core Functionality

---

Since we are offering a platform in which users can share their location-based stories, we have worked hard to allow users to tell their stories without excessively restricting them to the boundaries of available current locations. Rather we offer them to name any geolocation. Below you can find current status and the work of each team seperately.

## Frontend

---

**Status:** We have successfully implemented the location feature and integrated the Google Maps API using the “react-google-maps” wrapper library.

**Work:** We used the mapping functionality in two crucial parts of our program: create post page and discovery page.

For create post page we have added a small map from which registered users can scroll through and select numbers of geolocations. Registered users can then name these geolocations and delete them however they want to. This offers great flexibility for registered users who wants to share a post from an old location that does not exist anymore - or for those who just want to use a different (probably more meaningful) name for geolocation.

On the discovery page, registered users can scroll through all public posts available in the Postory and private posts from the registered users that they follow. Posts appear as markers on the map and when a registered user clicks on a marker, a little information box displaying the title and author of that post pops up. If registered users click on that information box, they will be redirected to the view page of the related post.

**Future Work:** Until the final customer meeting, we will implement the filtering mechanism into our application. It will allow users to filter the posts that they want to see (e.g., location name, location range). After implementing the mechanism, we will integrate this functionality into our discovery page and offer our users a much better experience.

### Related Issues:

- [#356](#)
- [#377](#)
- [#379](#)

### Related PRs:

- [#384](#)
- [#405](#)

## Android

---

**Status:** The users can choose multiple point-like locations for their posts and give a desired name to that geolocation depending on the memory they want to tell.

**Work:** Google Maps API is successfully integrated to Android application. Locations can be selected in a flexible manner.

Registered users can see the map on discovery page and there are markers on the locations that correspond to a post. When the user clicks on the marker, she is redirected to the page that allows interaction with the post. These interactions include commenting and liking.

At Create Post page, the user sees the map when she wants to pick locations for her story. Multiple point-like locations can be selected and the user can give them the name they want. Connections to backend are completed.

**Future Work:** The user should be able to filter posts according to time, location etc. These features will be added to the discovery page. Moreover, we will try to implement a timeline of posts around a location. This will require some additions to mapping functionality. The frontend teams shows some popups at discovery page when the user clicks on a marker, but Android team did not implement it. The design of frontend is better, the Android team will also try to fix this deficiency.

#### Related Issues:

- [#374](#)
- [#376](#)
- [#410](#)
- [#413](#)

#### Related PRs:

- [#375](#)
- [#412](#)
- [#414](#)

## Backend

---

**Status:** After users choose locations, all of the locations are sent to the backend. They are sorted as list of location objects. Also, they are stored in order.

**Work:** Frontend and android sent the data as in the form of we want. The locations object are created and inserted to the post objects. They are stored in the order of frontend and android teams sent.

**Future Work:** The filtering and searching mechanisms will be added. New API functionalities will be implemented for this. This filtering and searching mechanisms will be used in discovery page. The form of data will be discussed and determined with frontend and android team.

#### Related Issues:

- [#397](#)

#### Related Pull Request:

- [#396](#)

# Satisfaction of Requirements

---

This part of the report aims to evaluate the satisfaction of project requirements in the second milestone. The satisfaction status of project requirements will be classified into three classes:

- 1 - Satisfied (Completed)
- 2 - Partially satisfied (In progress)
- 3 - Unaddressed (Not completed)

## Summary

---

Until the first milestone, the team had tried to implement the core functionalities of our application. On the other hand, this was just an oversimplified prototype of the project. The implementation made until Milestone 1 lacked three main features: authentication, mapping functionality and time resolution options for posts.

Without authentication, everybody could create, edit or view any post with dynamic nicknames. Until Milestone 2, we have completed the authentication feature. Thus, users now have data that allow the system to identify each person. The data includes nicknames, mail addresses and passwords. It is possible for the system to send emails to the users. Moreover, owner of each post is now precise, so not everybody can edit every post.

The absence of mapping feature weakened the meaning and purpose of our application before. Now, this feature is also implemented and the app is much more useful. This feature is also a prerequisite for searching and filtering based on location. It was very important that we completed it until the 2nd Milestone.

Time resolution options for posts was also another core functionality of our project. People may not always remember exact dates of their memories or exact dates might be redundant in some cases. On the other hand, sometimes even minutes are important when people tell a story. This feature is one of the aspects that make our project special and it is realized now.

Many other features other than these are also implemented. Our project still needs to satisfy many requirements, but it would not be too bold to say we made good progress. The assessment of all requirements is listed below.

## List of addressed requirements

---

This part contains the project requirements that the team tried to address until Milestone 2. Explanations to partially satisfied requirements will be numbered in square brackets. At the end of partially satisfied requirements list, you can check the reasons why the requirements were not completely satisfied(in progress). Below are the lists of "requirements in progress" and "completed" requirements:

### Requirements in Progress

- **1.1.1.1.1** Registered Users shall be able to fill their profile page with their information. This information includes their names, surnames, profile photos.[2]
- **1.1.1.1.5** Registered Users shall be able to search posts.[4]
- **1.1.1.3.1** Guests shall be able to view the main page.[3]
- **1.1.1.3.2** Guests shall be able to view public posts.[3]
- **1.1.1.3.3** Guests shall be able to search posts.[3][4]
- **1.1.2.3.1** Guests shall be able to enter the web application without signing up.[3]
- **1.1.2.5.4** Registered users shall be able to view the posts from registered users they follow or posts that contain the places/tags they follow on the registered user's main page.[1]

- **1.1.2.10.1** Registered users shall be able to edit their bios, names, surnames.[2]

## Explanations

1. Following places or tags is not possible yet, users can be followed.
2. Users can only change their profile photos at the moment.
3. Guests can only sign up at the moment. But these functionalities are already implemented for Registered users and after the necessary modifications they can be implemented for the Guests as well.
4. Users can search through the posts using the map on the discover page. They can not yet filter the posts.

## Completed Requirements

- **1.1.1.1.2** Registered Users shall be able to view posts.
- **1.1.1.1.3** Registered Users shall be able to sign in by using their username or email, and password.
- **1.1.1.1.4** Registered Users shall be able to sign out.
- **1.1.1.1.7** Registered Users shall be able to edit their own posts.
- **1.1.1.3.6** Guests shall be able to sign up.
- **1.1.2.1.1** Guests shall be able to sign up via providing their name, surname and e-mail address, and choosing a username and password.
- **1.1.2.2.1** Registered users shall be able to sign in with their username/e-mail address and password.
- **1.1.2.4.1** Registered users shall be able to create posts by selecting a location or multiple locations.
- **1.1.2.4.3** Registered users shall be able to add pictures to their posts.
- **1.1.2.4.4** Registered users should add date and time to their posts at the desired specific level as they remember their memories.
- **1.1.2.4.6** Registered users shall be able to tag their posts with words related to their posts.
- **1.1.2.6.1** Registered users shall be able to search for posts using the map.
- **1.1.2.7.1** Registered users shall be able to leave comments under the posts.
- **1.1.2.7.2** Registered users shall be able to like posts of other registered users that they follow or that have public profiles.
- **1.1.2.9.1** Registered users shall be able to follow the registered users with public profiles without sending request.
- **1.1.2.13.1** Registered users shall be able to edit the main part of their posts.
- **1.1.2.13.2** Registered users shall be able to edit the tags of their posts.
- **1.1.2.13.3** Registered users shall be able to edit the locations on their posts.
- **1.2.2.1** System shall have a "Forgotten Password" option in case registered users forget their password. Registered users will enter their emails and a link where they can change their password will be sent by the system via their emails.
- **2.2.1.1** Registered user passwords should be at least 8 characters long and should contain at least 1 uppercase letter, 1 number, 1 symbol.
- **2.2.1.2** Registered user passwords should not contain the registered user's username.
- **2.2.2** A verification email should be sent to the registered user if password change is requested. Until the registered user verifies the change, the password should not be changed.
- **2.2.4.1** Until the verification is done, the registered user can't enter the app.
- **2.3.1** Registered users' sign up data shall be stored in database.
- **2.3.4** Registered users' posts shall be stored in database.

- **2.4.1** There should be a web interface for the project.
- **2.4.1.1** Known web browsers(such as Chrome, Opera, Firefox, Microsoft Edge, Safari) should support the web site.
- **2.4.2** There should be an android-based mobile app for the project.
- **2.5.1.1** Response time shall be short enough not to harm interactivity.
- **2.5.1.2** Reducing the response time for users should be prioritized.

## List of unaddressed requirements

---

This part includes requirements that were not addressed by the team. Clearly, all of these requirements are not completed yet. Below are the lists of uncompleted functional and non-functional requirements:

### Unaddressed Functional Requirements

- **1.1.1.1.6** Registered Users shall be able to filter posts.
- **1.1.1.1.8** Registered Users shall be able to save posts.
- **1.1.1.2.1** Admins shall be able to view posts.
- **1.1.1.2.2** Admins shall be able to revise reports of inappropriate post.
- **1.1.1.2.3** Admins shall be able to ban registered users.
- **1.1.1.2.4** Admins shall be able to remove a post.
- **1.1.1.2.5** Admins shall be able to remove a comment under a post.
- **1.1.1.2.6** Admins shall be able to send mail to registered users.
- **1.1.1.3.4** Guests shall be able to filter posts.
- **1.1.1.3.5** Guests shall be asked to sign up for the rest of the registered user functionalities listed under 1.1.1.1.
- **1.1.2.4.2** Registered users shall be able to allow location services to find their current location and add a story to it.
- **1.1.2.4.5** Registered users shall be able to tag other people to their posts.
- **1.1.2.5.2** Registered users shall be able to view posts on the locations they choose from the map.
- **1.1.2.5.3** Registered users shall be able to view the number of views of a post.
- **1.1.2.6.1** Registered users shall be able to search for posts posted at a specific time and date.
- **1.1.2.7.3** Registered users shall be able to like comments under a post.
- **1.1.2.8.1** Registered users shall be able to save the posts in their own profiles to access later.
- **1.1.2.9.2** Registered users shall be able to send follow requests to private profiles.
- **1.1.2.9.3** Registered users shall be able to follow locations.
- **1.1.2.9.4** Registered users will be able to follow a private profile only if the owner of the private profile approves the follow request.
- **1.1.2.10.2** Registered users shall have a feed to display their activities.
- **1.1.2.10.3** Registered users shall be able to choose whether the profile page will be accessible to guests or registered users.
- **1.1.2.11.1** Registered users shall be able to choose to receive notifications or not.
- **1.1.2.12.1** Registered users shall be able to report inappropriate posts.
- **1.2.1.1** System shall ask the user to verify their address.
- **1.2.2.2** System shall provide the user a "Remember Me" option to save username and password in local.
- **1.2.3.1** System should request the guest users to sign up for them to use the Android application.
- **1.2.4.1** System shall notify admins in case of a report of an inappropriate post.
- **1.2.4.2** System shall notify registered users with private accounts when they receive a follow request.

- **1.2.4.3** System shall notify registered users when someone starts to follow them.
- **1.2.4.4** System shall notify registered users when someone likes their posts.
- **1.2.4.5** System shall notify registered users when someone comments on their posts.

## Unaddressed Non-Functional Requirements

- **2.2.3** Admins should change their passwords every 180 days. This should be reminded to the admin via email.
- **2.2.4** After 4 unsuccessful password entries, the registered users should verify that it was them by clicking the verification link sent to them via email.
- **2.2.4.2** After the verification, the registered users should reset their passwords.
- **2.5.2.1** Measures shall be taken to shorten the downtime as much as possible.
- **2.6.1** W3C Activity Streams 2.0 Protocol will be followed throughout the project implementation.
- **2.6.2** W3C Standards will be followed.

# API Endpoints

---

Endpoints are explained below. Example requests can be found in the following links:

- Follow users, get user information, get post by id examples can be found in this postman collection [link](#).
- Get main page posts, get discover page posts, get a specific user's posts, add profile photo to current user examples can be found in this postman collection [link](#).
- Sign up, activation, sign in, post create, post update, comment and like examples can be found in this postman collection [link](#).

## Sign Up

---

User account can be created by using this endpoint. User credentials will be sent and user information will be returned.

**Link:** POST <http://3.67.83.253:8000/auth/users/>

## Activation

---

User accounts can be activated by using this endpoint. Token will be sent to user's email.

**Link:** POST <http://3.67.83.253:8000/auth/users/activation/>

## Sign In

---

Users can sign in to the application. JWT tokens will be returned for session management in the application.

**Link:** POST <http://3.67.83.253:8000/auth/jwt/create/>

## Post Create

---

A post will be created. Post fields will be sent. The created post will be returned. It will also added to the user's post lists. Authorization field in the header is required. JWT token should be added to this field.

**Link:** POST <http://3.67.83.253:8000/api/post/create>

## Post Update

---

Posts will be updated using this endpoint. New fields will be sent and updated post will be returned. Authorization field in the header is required. JWT token should be added to this field.

**Link:** PUT [http://3.125.114.231:8000/api/post/put/\(int:story-id\)](http://3.125.114.231:8000/api/post/put/(int:story-id))

## Main Page Posts

---

For the main page posts which includes followed user's of the current user and own posts of the current user this API endpoint is used. Authorization field in the header is required. JWT token should be added to this field.

**Link:** GET  
<http://3.67.83.253:8000/api/post/all>

## Discover Page Posts

---

For the discover page posts which includes followed user's of the current user and posts of public users this API endpoint is used. Authorization field in the header is required. JWT token should be added to this field.

**Link:** GET

<http://3.67.83.253:8000/api/post/discover>

## Specific User's Posts

---

In a specific users profile page, their posts should be shown. To get their posts this API endpoint is used. Authorization field in the header is required. JWT token should be added to this field.

**Link:** GET

[http://3.67.83.253:8000/api/post/user/\(int:user-id\)](http://3.67.83.253:8000/api/post/user/(int:user-id))

## Profile photo

---

To update a user's profile photo, this endpoint is used. File is sent in form-data format. JWT token should be added to this field.

**Link:** POST

<http://3.67.83.253:8000/api/user/addPhoto>

## Comment

---

Users can leave comments to the posts by using the story IDs. Story will be returned. Authorization field in the header is required. JWT token should be added to this field.

**Link:** POST [http://3.125.114.231:8000/api/post/comment/\(int:story-id\)](http://3.125.114.231:8000/api/post/comment/(int:story-id))

## Like

---

Users can like or unlike by using the story IDs. Story will be returned. Authorization field in the header is required. JWT token should be added to this field. This does not have a payload.

**Link:** POST [http://3.125.114.231:8000/api/post/like/\(int:story-id\)](http://3.125.114.231:8000/api/post/like/(int:story-id))

## Follow User

---

Users can follow each other. The requirements for the follow is that the followed user should have a public account. Moreover, the user can't follow a user twice or he/she can't follow himself/herself. Authorization field in the header is required. JWT token should be added to this field.

- If the user tries to follow itself: *HTTP\_400\_BAD\_REQUEST*
- If the user tries to follow another user again: *HTTP\_409\_CONFLICT*
- If the user tries to follow a private user: *HTTP\_403\_FORBIDDEN*

**Link:** POST [http://3.125.114.231:8000/api/user/follow/\(int:user-id\)](http://3.125.114.231:8000/api/user/follow/(int:user-id))

## View a Single Post

---

Users can see posts of the users they follow or their own posts. The post will be returned. Authorization field in the header is required. JWT token should be added to this field. If the user is not allowed to view the post, *HTTP\_401\_UNAUTHORIZED* will be returned.

**Link:** GET [http://3.125.114.231:8000/api/post/get/\(int:story-id\)](http://3.125.114.231:8000/api/post/get/(int:story-id))

## **View a Single User**

Users can see other users information. This is allowed when the user itself tries to view or he/she tries to view a followed user's information.

**Link:** GET [http://3.125.114.231:8000/api/user/get/\(int:user-id\)](http://3.125.114.231:8000/api/user/get/(int:user-id))

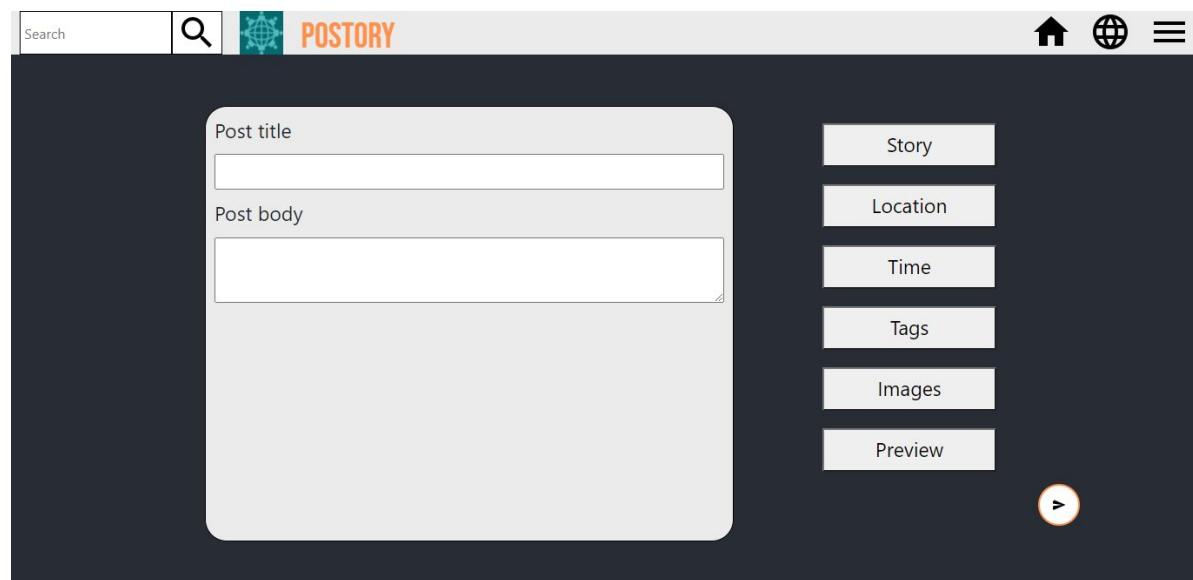
## **User Experience Pages For The Web**

### **Screen shots for create/edit post pages**

#### **Link to code**

<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/CreatePost.js>

#### **Adding Story**



# Adding Time

The screenshot shows the POSTORY interface with a dark theme. At the top left is a search bar with a magnifying glass icon. Next to it is the POSTORY logo, which includes a globe icon and the word "POSTORY". On the top right are icons for home, globe, and menu.

The main area contains three sets of input fields for selecting time periods:

- Choose start and end years: Input fields for "1999" and "2000", and a "Clear" button.
- Choose start and end months: Input fields for "1" and "12", and a "Clear" button.
- Choose start and end days: Input fields for two dates, and a "Clear" button.

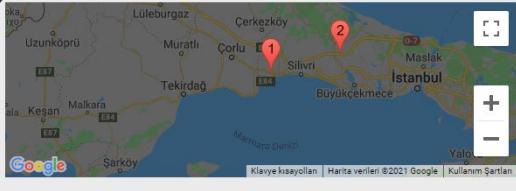
To the right of these input fields is a vertical column of buttons:

- Story
- Location
- Time
- Tags
- Images
- Preview

A small orange circular arrow icon with a white triangle is located at the bottom right of the screen.

# Adding Locations

Search   POSTORY   



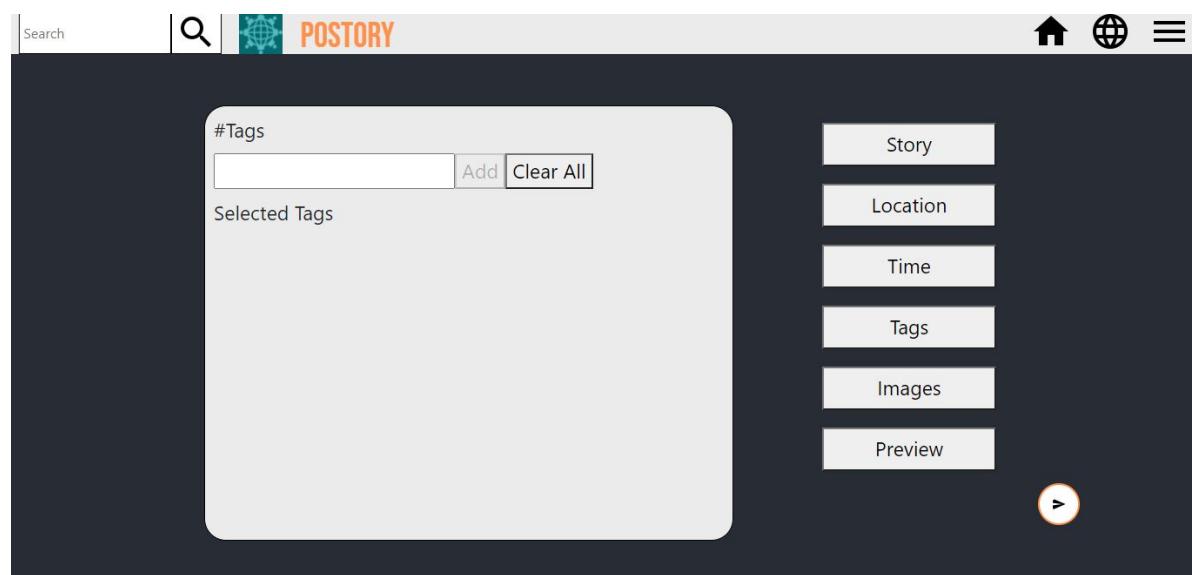
Please use below if you want to add names to the locations.

- 1 location1
- 2 location2

Story  
Location  
Time  
Tags  
Images  
Preview



# Adding Tags



## Seeing the preview

The screenshot shows the POSTORY mobile application interface. At the top, there is a search bar with a magnifying glass icon, followed by the POSTORY logo. On the right side of the top bar are icons for home, globe, and menu.

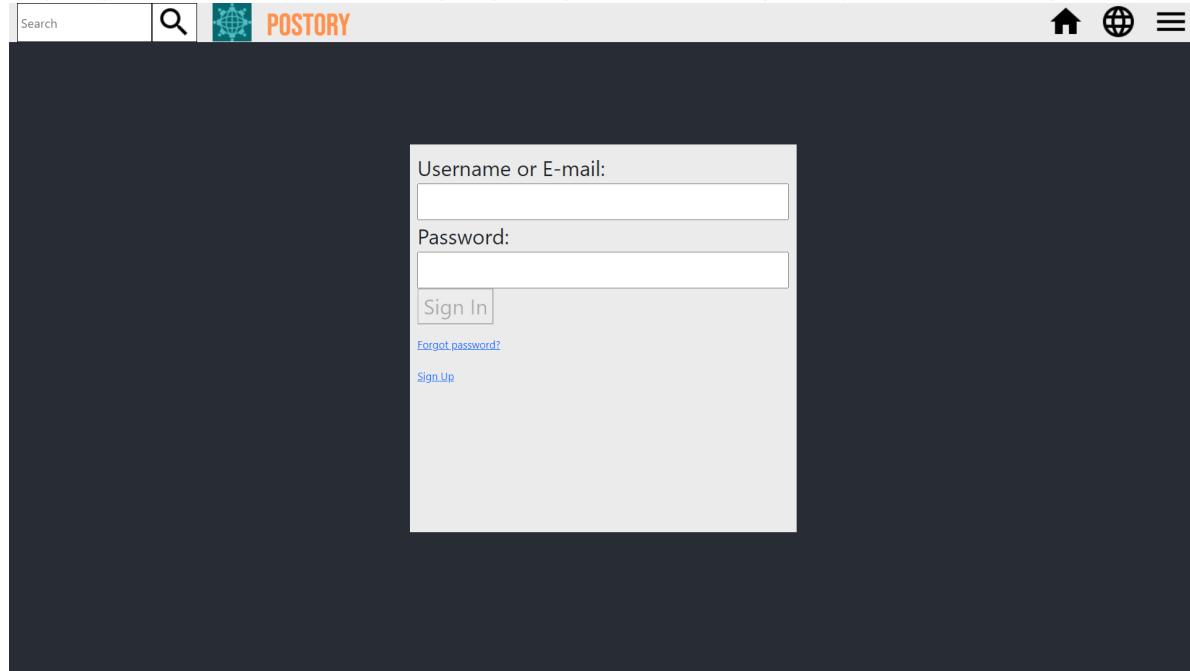
The main content area displays a post card. The post card features a profile picture of a person with the text "ME" next to it. Below the profile picture is a small red button with the word "Food". The title of the post is "Test". The post content reads: "Two months ago I was in Konya, and I ate etli ekmek in Havzan 2. It was very delicious and I liked it a lot. I recommend everybody to go there and take a visit." Below the text is a thumbnail image of a man's face. At the bottom of the post card are several interaction icons: a heart with a zero, a speech bubble, a share symbol, a bookmark, and an exclamation mark.

To the right of the post card is a vertical sidebar with the following options: Story, Location, Time, Tags, Images, and Preview. A circular arrow icon is located at the bottom right of the sidebar.

# Sign-In page

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/SignIn.js>



# Sign-Up page

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/SignUp.js>

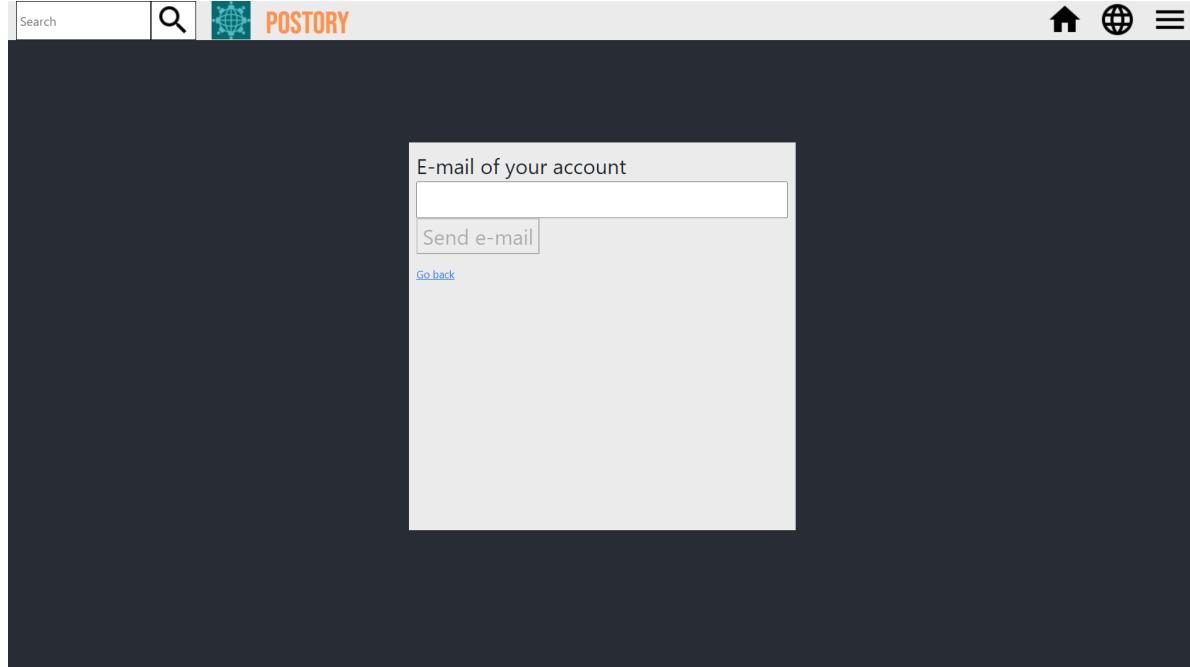
The screenshot shows a dark-themed web application interface. At the top left is a search bar with a magnifying glass icon. Next to it is a logo consisting of a globe icon and the word "POSTORY". On the far right of the header are icons for home, refresh, and menu. The main content area contains a form for user registration. The form fields are labeled "E-mail:", "Username:", "Name:", "Surname:", "Password:", and "Repeat password:". Below the "Repeat password:" field is a "Sign Up" button. Underneath the "Sign Up" button is a small link "Sign in instead".

E-mail:	<input type="text"/>
Username:	<input type="text"/>
Name:	<input type="text"/>
Surname:	<input type="text"/>
Password:	<input type="password"/>
Repeat password:	<input type="password"/>
<input type="button" value="Sign Up"/>	
<small><a href="#">Sign in instead</a></small>	

# 'Forgot password' page

## Link to code

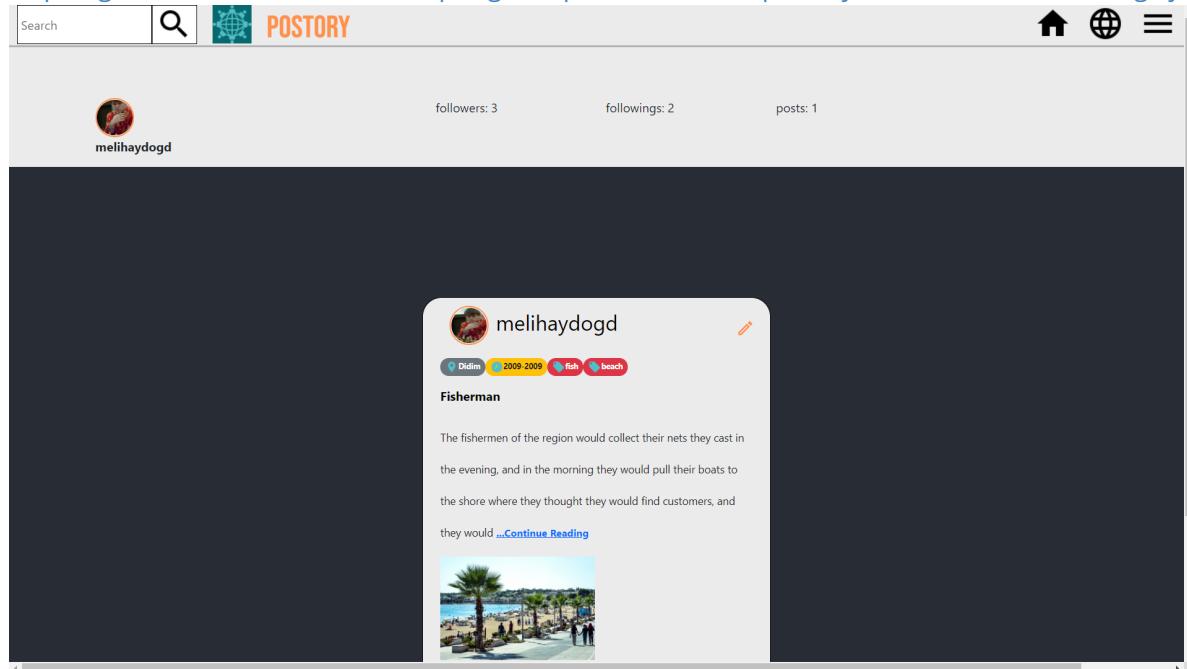
<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/ForgotPassword.js>



# Profile page

## Link to code

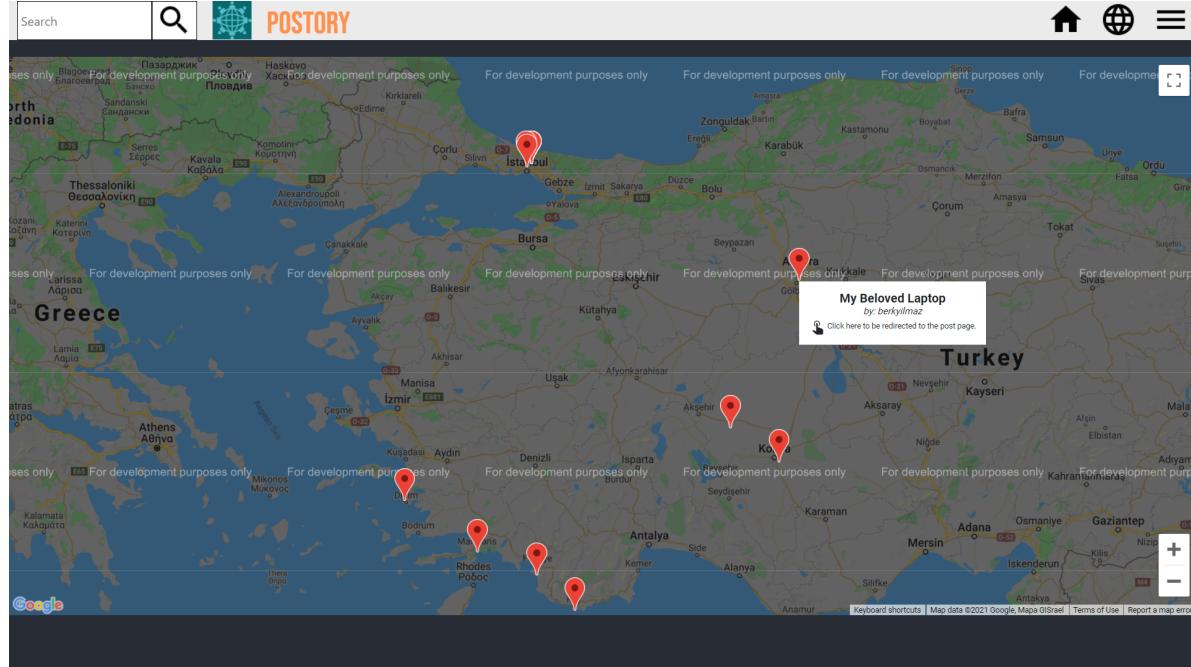
<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/ProfilePage.js>



# Discovery page

## Link to code

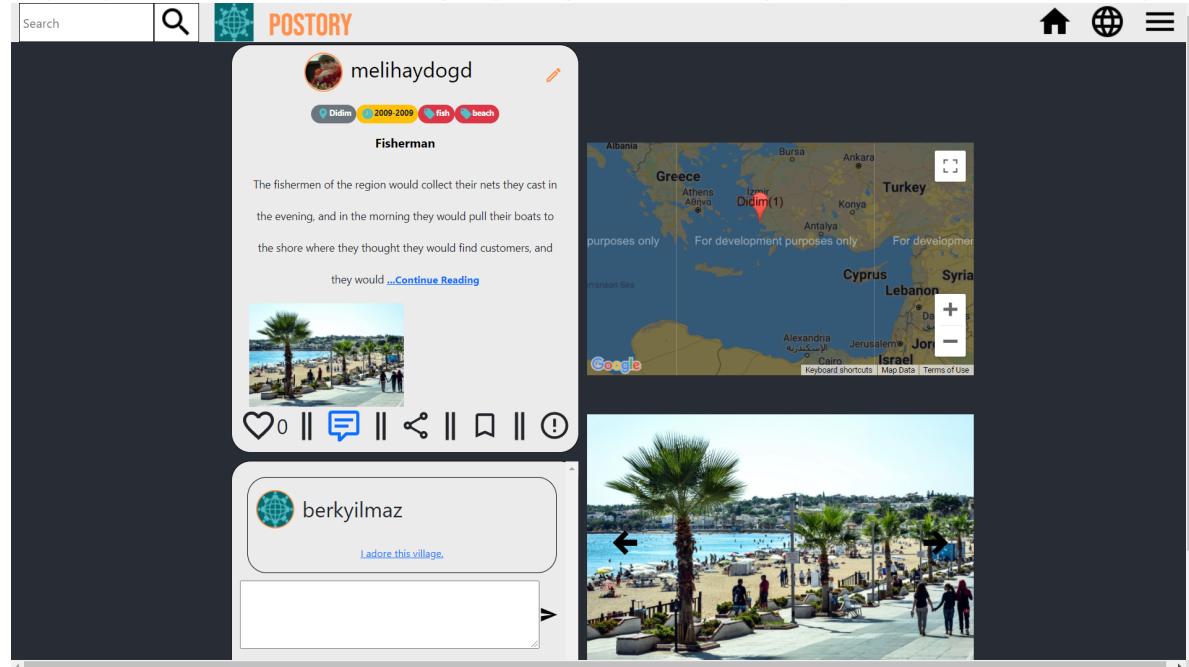
<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/DiscoverPage.js>



# Viewing posts page

## Link to code

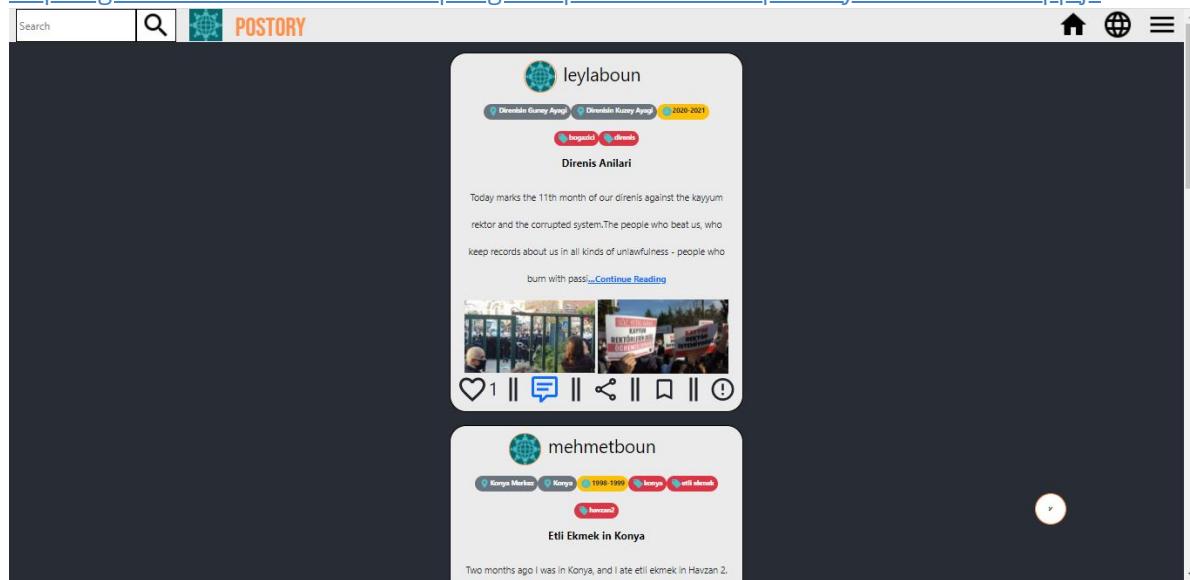
<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/ViewPost.js>



# Main page

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/master/postory/frontend/src/App.js>



# User Experience Pages For The Mobile Application

## Login Page

**Link to code**

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/app/src/main/java/com/example/postory/activities/LoginActivity.java>



Write your mail address

Write your password



Sign in

Forgot password?

---

Sign Up

# Signup Page

---

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/RegisterActivity.java>



Write your name

---

Write your surname

---

Write your mail address

---

Write your username

---

Write your password

---



Write your password again

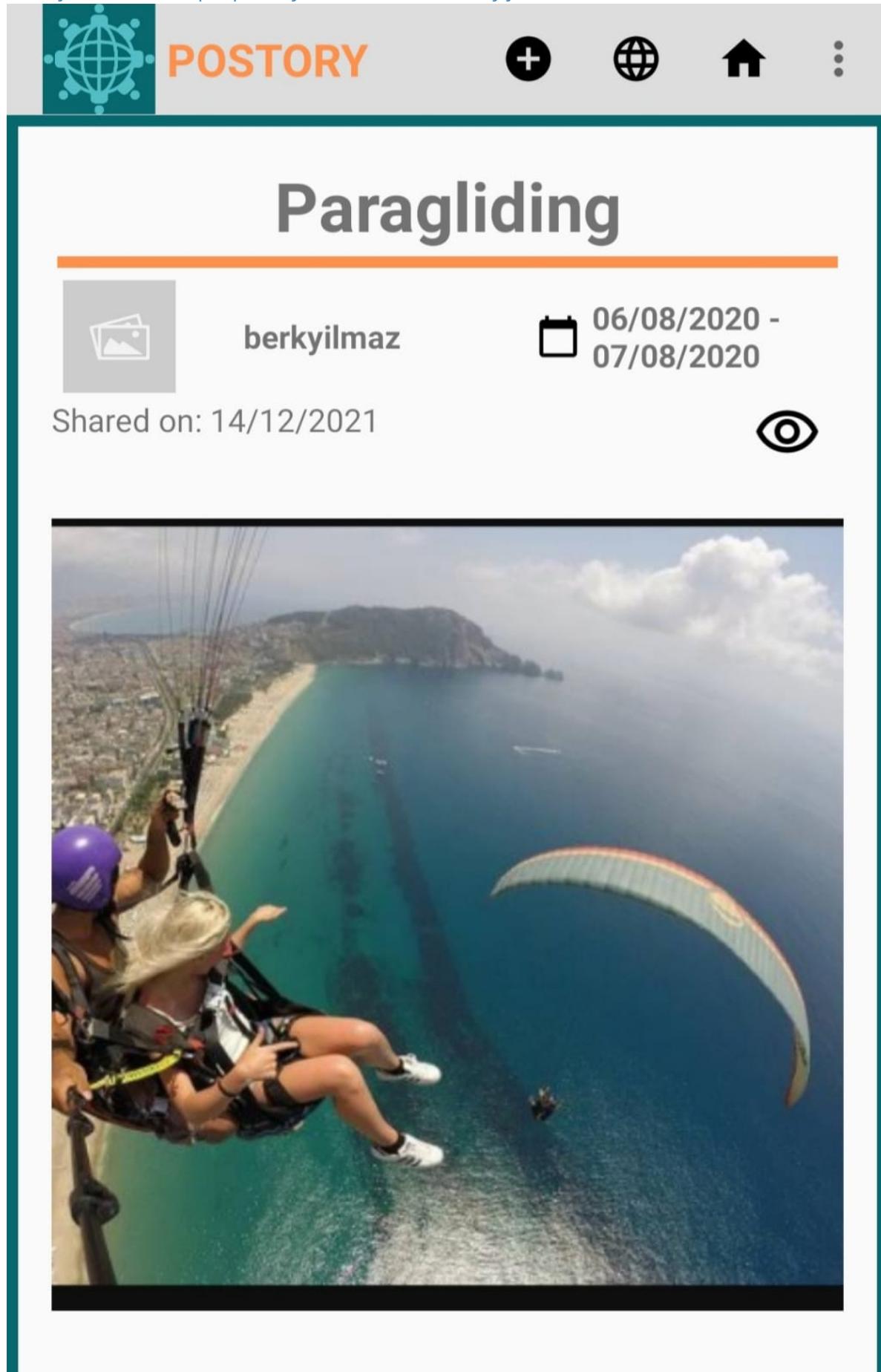


**Sign Up**

# Main Page

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/MainActivity.java>



#fun #adrenaline

 Paraglide

I really enjoy paragliding and on this day, this was my very first time as a paragliding instructor. I really look forward for the next summer.

## Profile Page

### Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/SelfProfilePageActivity.java>

The screenshot shows the profile page of a user named 'berkyilmaz'. At the top, there is a navigation bar with a globe icon, the app name 'POSTORY' in orange, and various icons for adding posts, viewing the feed, and more. Below this, the user's information is displayed: Username 'berkyilmaz', Name 'Berk', and Surname 'Yılmaz'. To the left of the profile info is a placeholder image for a profile picture. Below this section, there are statistics: 'Followed by: 1', 'Following: 2', and 'Number of posts: 2'. A large 'POSTS' heading is centered below these stats. The first post is titled 'Paragliding' and features a thumbnail image of a paraglider in flight over a coastal town and beach. The post details include the author 'berkyilmaz', the date range '06/08/2020 - 07/08/2020', and the sharing date 'Shared on: 14/12/2021'. There is also a view count icon.

Username: berkyilmaz

Name: Berk

Surname: Yılmaz

Followed by: 1

Following: 2

Number of posts: 2

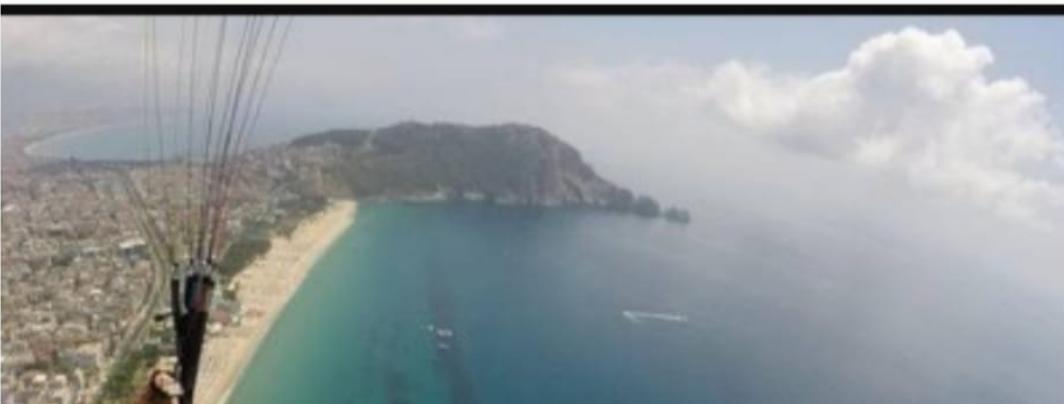
## POSTS

### Paragliding

berkyilmaz

06/08/2020 - 07/08/2020

Shared on: 14/12/2021

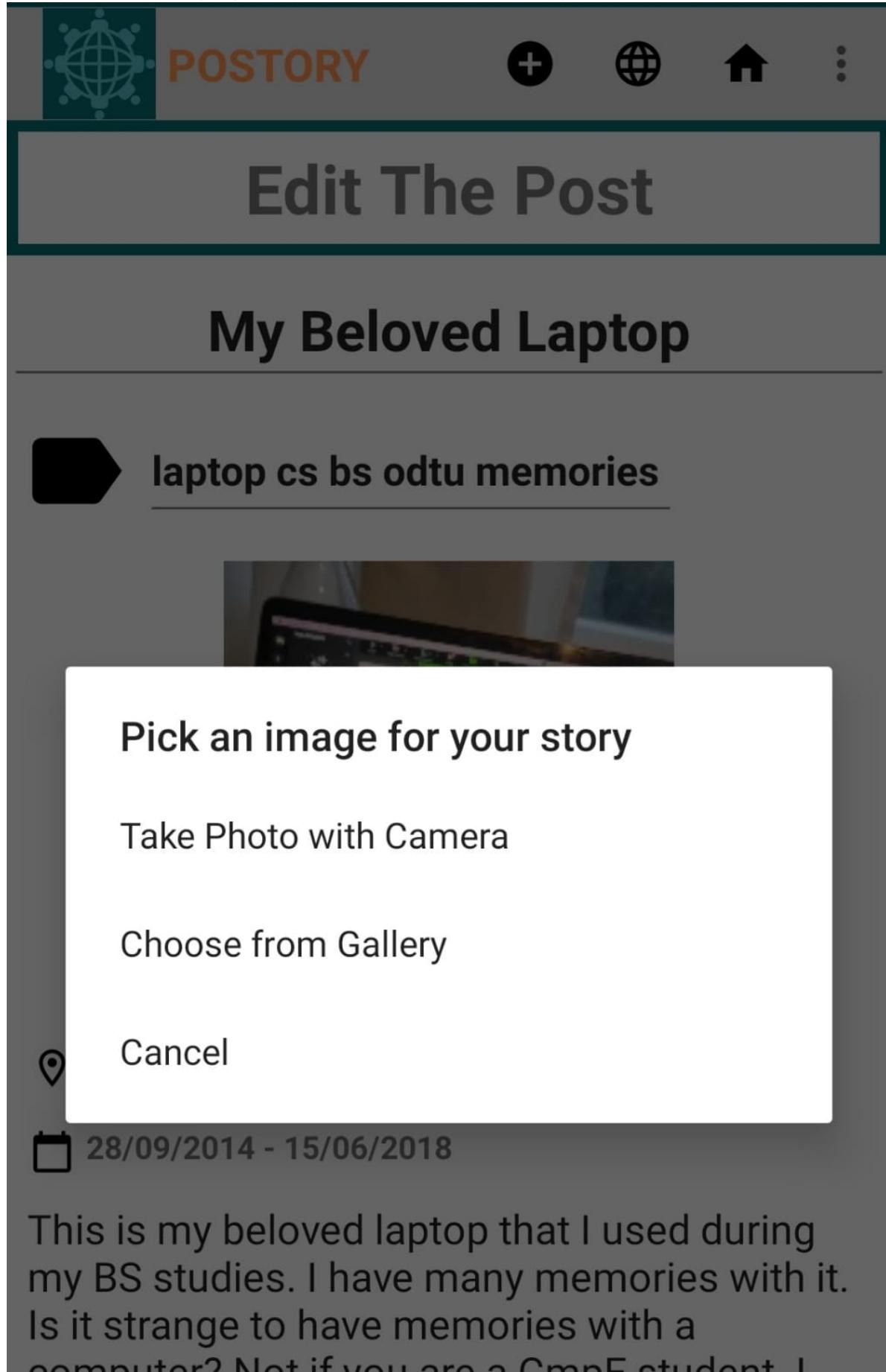




## Pick Image

[Link to code](#)

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/CreatePostActivity.java>



computer. Now you are a simple student. I remember so many nights that I worked hard to catch deadlines. This laptop always supported me. It is still in good shape and I use it. its still in very good shape.

---

SEND POST

## Edit Posts

### Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/CreatePostActivity.java>

**POSTORY**

**Edit The Post**

# My Beloved Laptop

laptop cs bs odtu memories



📍 Home Odtü

📅 28/09/2014 - 15/06/2018

This is my beloved laptop that I used during my BS studies. I have many memories with it. Is it strange to have memories with a computer? Not if you are a CmpF student. I

remember so many nights that I worked hard to catch deadlines. This laptop always supported me. It is still in good shape and I use it. its still in very good shape.

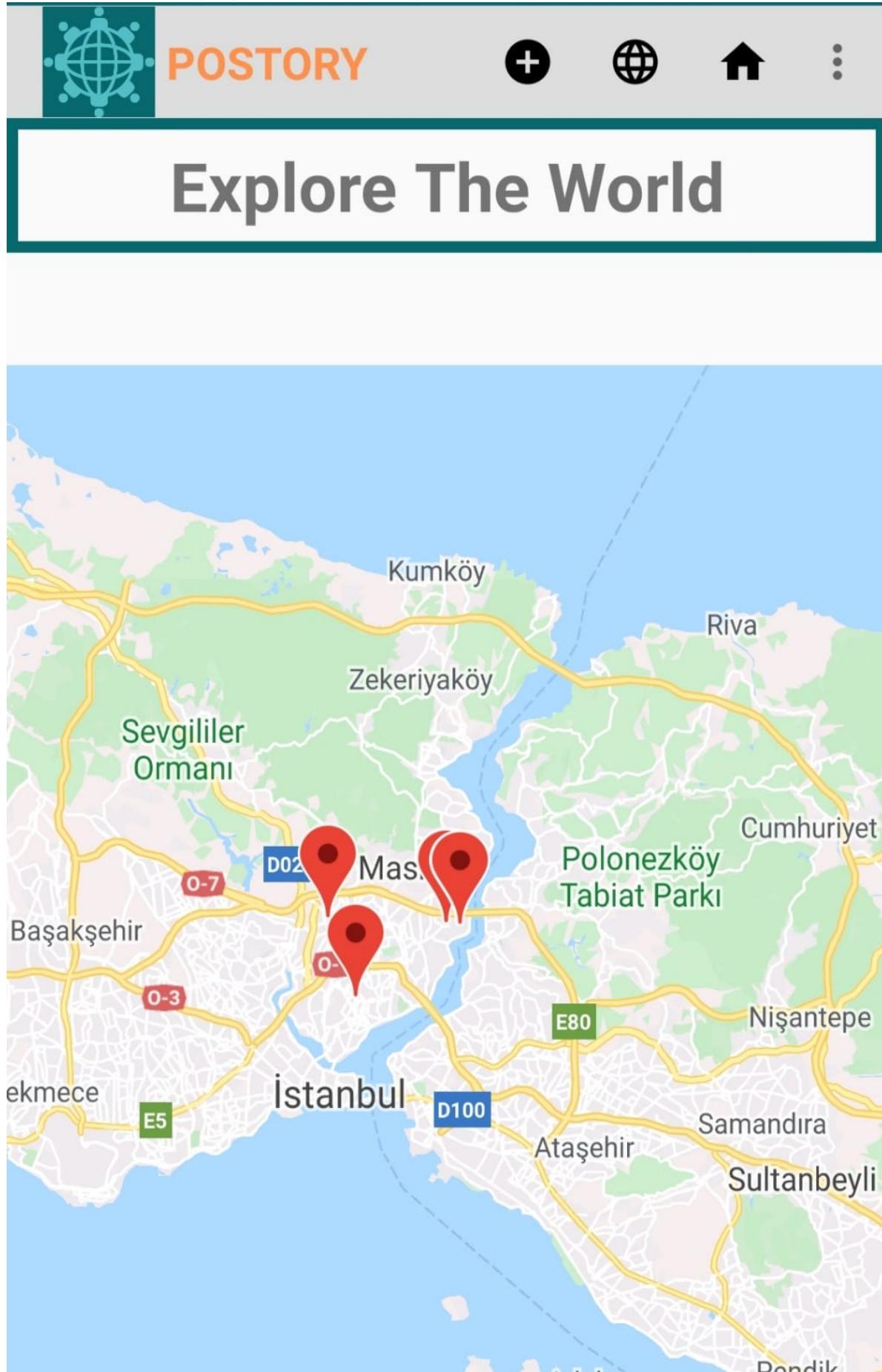
---

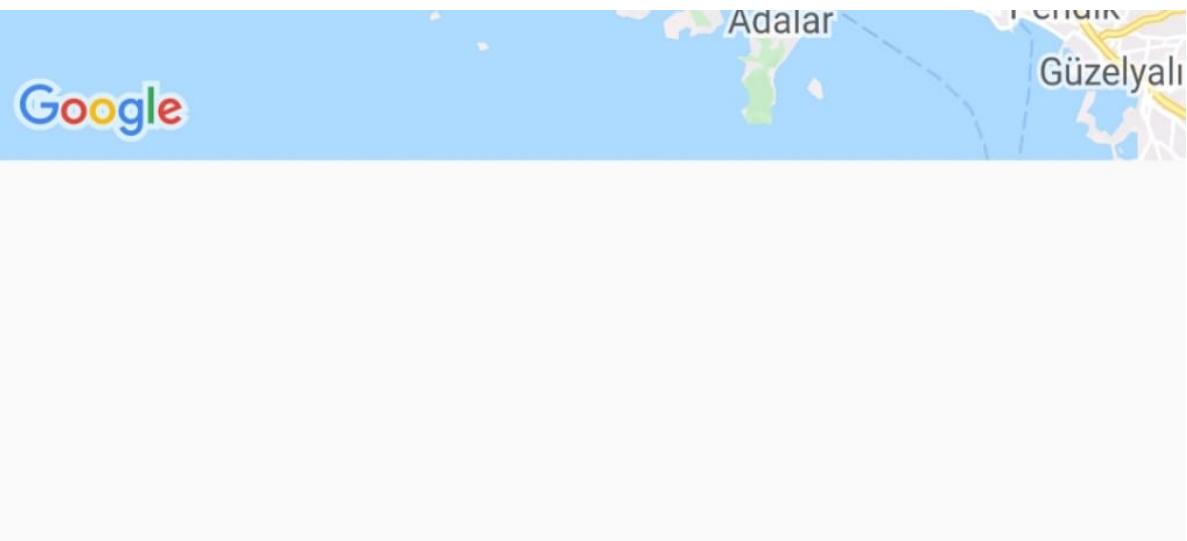
SEND POST

## Explore Page

[Link to code](#)

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/ExploreActivity.java>





## Create Post

### Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/CreatePostActivity.java>

The screenshot shows the 'Create New Post' screen of the Postory app. At the top, there is a navigation bar with a globe icon, the word 'POSTORY' in orange, and three icons: a plus sign, a globe, a house, and a vertical ellipsis. Below the navigation bar is a large title 'Create New Post'. A text input field is labeled 'Write a Title...'. To the left of the text input is a black arrow pointing right, followed by the instruction 'Enter Tags One Space Apart'. Below the text input is a large black button featuring a white camera icon with a plus sign above it. At the bottom left, there is a list of two items with icons: a pin for locations and a calendar for time. At the very bottom is a blue button labeled 'SEND POST'.

- Click the pin on the left to enter multiple locations
- Click on the calendar to the left to tell the time of your story.

Tell the Story...

SEND POST



# Single Post Page

## Link to code

<https://github.com/bounswe/2021SpringGroup9/blob/android/postory/android/Postory/app/src/main/java/com/example/postory/activities/SinglePostActivity.java>



#bogazici #direnis

📍 Bogazici Kuzey



Like

Comment

2 people liked this.

We are experiencing, most probably, the most interesting times at Bogazici University. For almost 1 year we are protesting against the kayyum rector and his ideology. All students, teachers - all components of the Bogazici University with sincerity rebel against this undemocratic movement. We will be here always.  
#kabuletmiyoruzvazgeçmiyoruz

[Continue Reading](#)

# Comments

---

leylaboun #kabuletmiyoruzvazgeçmiyoruz

---

berkyilmaz #boun

## Work Done by Each Team Member

---

Name	Work Done
<i>Mert Alkan</i>	<ul style="list-style-type: none"> <li>• Researched about authentication on Django</li> <li>• Learned about Djoser as the authentication provider.</li> <li>• Created endpoint for getting main page posts</li> <li>• Created endpoint for getting discover page posts</li> <li>• Created endpoint for getting specific user's posts</li> <li>• Fixed a bug on getting a specific post</li> <li>• Fixed a bug on retrieving posts and returning username aswell</li> <li>• Fixed a bug on retrieving user.</li> <li>• Fixed a bug on creating a post.</li> <li>• Fixed a bug on creating a post.</li> <li>• Changed user profile photos to be on only user model.</li> <li>• Fixed a bug on creating second user.</li> <li>• Wrote milestone introduction, personal milestone.</li> <li>• Creted PRs <a href="#">#383</a>,<a href="#">#393</a>,<a href="#">#411</a>,<a href="#">#417</a>,<a href="#">#436</a>,<a href="#">#438</a></li> <li>• Reviewed PRs <a href="#">#365</a>,<a href="#">#385</a>,<a href="#">#386</a>,<a href="#">#387</a>,<a href="#">#394</a>,<a href="#">#396</a></li> </ul>
<i>Ahmet Melih Aydoğdu</i>	<ul style="list-style-type: none"> <li>• Attended weekly meetings and team meetings</li> <li>• Researched authentication systems for backend</li> <li>• Added Djoser settings to the backend project</li> <li>• Updating user model</li> <li>• Adding user photo functionality</li> <li>• Added comment and like functionalities</li> <li>• Added updated location and time fields for post creation</li> <li>• Created new EC2 instance and S3 bucket because my AWS account suspended</li> <li>• Updated IP addresses for new EC2 instance</li> <li>• Updated corsheaders for authentication</li> <li>• Worked on issues: #369, #370, #397</li> <li>• Opened pull requests: #365, #385, #396, #464</li> <li>• Wrote reviews to pull requests: #383, #386, #387, #393, #394, #438</li> <li>• Wrote the individual report for Milestone 2</li> </ul>

Name	Work Done
Zehranaz Canfes	<ul style="list-style-type: none"> <li>• Make a Research for Authentication</li> <li>• Watch video for JWT/Djoser Authentication system</li> <li>• Read the documentation of djoser</li> <li>• Read the documentation to find a way to get user id</li> <li>• Make a research for session storage of signed in user</li> <li>• Try to use JWT and fix the bugs occurred during it. I saw that the requirements.txt was incompatible with the others. I used other requirements for the code to work. e.g. I added sqlparse and a different PyJWT version.</li> <li>• Implemented the user follow functionality. I encountered many bugs during the implementation. That's why it took a lot of my time. <a href="#">Issue 368</a></li> <li>• Implemented get user by id functionality. <a href="#">Issue 378</a></li> <li>• Add authentication control for user follow and get user by id functionality. Only signed up users can use these functionalities</li> <li>• <a href="#">Extend "get post by id" functionality for checking if user has access to other user's posts #373</a></li> <li>• <a href="#">Add user_id check to update post #388</a></li> <li>• Written Two Posts for Postory App</li> <li>• Attended all meetings</li> <li>• Attended all labs</li> <li>• Wrote the Moving Forward part of the Executive summary of this report</li> <li>• Wrote individual report</li> <li>• Wrote weekly efforts</li> <li>• The links to PRs I opened and the reviews I wrote can be found in my individual report.</li> <li>• The PRs reviewed by me: <a href="#">PR 365</a>, <a href="#">PR 383</a>, <a href="#">PR 385</a>, <a href="#">PR 393</a>, <a href="#">PR 396</a>, <a href="#">PR 483</a></li> <li>• The PRs opened by me: <a href="#">PR #387</a>, <a href="#">PR #386</a>, <a href="#">PR #394</a>,</li> <li>• Wrote the API documentation for user follow, get user, get post by id which can be found in this report.</li> </ul>

Name	Work Done
Mustafa Emir Çolak	<ul style="list-style-type: none"> <li>Implemented the view post page (shows the more detailed information of a post).</li> <li>Implemented location choosing using google maps in the create post page (multiple and named locations can be created).</li> <li>Held a meeting with the frontend team to integrate the frontend with the backend API. After the meeting I continued to integrate some of the other components.(Create/edit post, view post)</li> <li>Researched about the react testing library, Added a test for the post component on 5013327.</li> <li>React router link on discovery page was crashing the application. Changed it with a redirect on clicking the button.</li> <li>The user can now add a profile photo on their profile page after clicking their profile picture.(This functionality will probably changed for a settings page in the future.</li> <li>Attended all of the meetings I needed to be in. Took the meeting notes for <a href="#">lab 7</a> and frontend <a href="#">meeting 3</a>.</li> <li>Listing part of the status of requirements in three groups for the group milestone report 2.</li> </ul>
Ömer Barış Erkek	<ul style="list-style-type: none"> <li>Attended Discord meetings of our group</li> <li>Attended meetings of the frontend team</li> <li>Implemented my works in separate branches, created and managed issues for them, and opened pull requests for them when the work is completed</li> <li>Tested and reviewed pull requests of my teammates</li> <li>Researched on djoser, JWT tokens, and how to implement them in the frontend</li> <li>Implemented authentication related React components for the frontend as well as their backend integrations: SignUp, SignIn, Activation, ForgotPassword, and ForgotPasswordConfirm</li> <li>Improved TimeChooser component to better choose date and time</li> <li>Edited TimeChooser and EditPost components so that TimeChooser could be edited</li> <li>Fixed several errors and bugs in my components</li> <li>Helped the frontend team complete the backend integrations</li> <li>Written my Milestone 2 Personal Report</li> <li>Contributed to the Milestone 2 Group Report</li> </ul>

Name	Work Done
<i>Melih Özcan</i>	<ul style="list-style-type: none"> <li>• Attended the group and LAB meetings</li> <li>• Implemented the explore page</li> <li>• Integrated the Google Maps SDK</li> <li>• Implemented the explore page</li> <li>• Implemented the like functionality</li> <li>• Implemented the comment functionality</li> <li>• Implemented the take photo functionality</li> <li>• Attended Lab Meetings 6,7, 8</li> <li>• Prepared the individual report</li> <li>• Contributed to the Group Report</li> <li>• Created the issues 374, 376, 410, 413, 420 422, 443, 445,447 and 459</li> <li>• Created the pull requests 375, 412, 414, 421, 423, 442, 453, 456, 457, 458</li> </ul>
<i>Ahmet İbrahim Şentürk</i>	<ul style="list-style-type: none"> <li>• Implemented discovery page</li> <li>• Implemented profile page</li> <li>• Research Map API</li> <li>• Research Front-end Framework</li> <li>• Implemented like post mechanism</li> <li>• Fixed the layout of frontend</li> <li>• Took meeting notes</li> <li>• Explained mapping functionality in frontend for milestone 2 report</li> <li>• Wrote a guideline for the milestone 2 presentation</li> <li>• Worked on issues: <a href="#">#353</a>, <a href="#">#356</a>, <a href="#">#379</a>, <a href="#">#392</a>, <a href="#">#425</a>, <a href="#">#426</a>, <a href="#">#435</a>, <a href="#">#448</a>, <a href="#">#449</a>, <a href="#">#475</a>, <a href="#">#391</a></li> <li>• Opened PRs: <a href="#">#384</a>, <a href="#">#418</a>, <a href="#">#434</a>, <a href="#">#437</a>, <a href="#">#450</a>, <a href="#">#452</a>,<a href="#">#462</a></li> <li>• Reviewed PRs: <a href="#">#371</a>, <a href="#">#372</a>, <a href="#">#389</a>, <a href="#">#405</a>, <a href="#">#407</a>, <a href="#">#408</a>, <a href="#">#419</a>, <a href="#">#428</a>, <a href="#">#432</a></li> <li>• Attended all of the meetings and lab sessions</li> <li>• Took meeting notes of <a href="#">Frontend Meeting 4</a></li> </ul>

Name	Work Done
Niyazi Ülke	<ul style="list-style-type: none"> <li>• Researched about Android development.</li> <li>• Researched about CI/CD, wrote Android CI/CD Pipeline and solved bugs in it.</li> <li>• Refactored previous Android code.</li> <li>• Wrote "Summary" of Satisfaction of Requirements part for this milestone report.</li> <li>• Explained mapping functionality implemented by Android team in this Milestone report.</li> <li>• Wrote individual milestone 2 report.</li> <li>• Took meeting notes at <a href="#">Android Meeting 2</a>.</li> <li>• Attended all meetings and labs.</li> <li>• Wrote TimeController and PasswordController classes that validate user time and password input.</li> <li>• Wrote tests for TimeController and PasswordController classes.</li> <li>• Implemented sign in, sign up and sign out functionalities for Android app and linked it to backend.</li> <li>• Implemented password reset functionality for Android.</li> <li>• Implemented direct login functionality for Android app that allows the user to login without reentering credentials from the same device.</li> <li>• Implemented follow a user functionality for Android.</li> <li>• Implemented profile page for Android application and connected it to backend.</li> <li>• Implemented time chooser for Android that allows users to choose start and end times of their posts with the desired precision as it is explained in requirements.</li> <li>• Worked on presentation scenario for customer presentation 2.</li> <li>• Worked on issues <a href="#">#351</a>, <a href="#">#360</a>, <a href="#">#361</a>, <a href="#">#398</a>, <a href="#">#424</a>, <a href="#">#446</a>, <a href="#">#466</a>, <a href="#">#467</a>, <a href="#">#468</a>, <a href="#">#469</a>, <a href="#">#470</a></li> <li>• Opened pull requests <a href="#">#350</a>, <a href="#">#359</a>, <a href="#">#403</a>, <a href="#">#409</a>, <a href="#">#415</a>, <a href="#">#416</a>, <a href="#">#427</a>, <a href="#">#433</a>, <a href="#">#451</a>, <a href="#">#455</a>, <a href="#">#460</a>, <a href="#">#461</a></li> <li>• Reviewed pull requests: <a href="#">PR 412</a>, <a href="#">PR 414</a>, <a href="#">PR 421</a>, <a href="#">PR 423</a>, <a href="#">PR 442</a>, <a href="#">PR 453</a>, <a href="#">PR 456</a>, <a href="#">PR 457</a>, <a href="#">PR 458</a></li> </ul>