



Final Report and Documentation

Team Members

Jaden Balogh.

Mathew de Vin.

Keyvan Khademi.

Carson Ricca

Project Description/Details

Overview

Postn is hosted at <https://postn-dev.herokuapp.com/>. While this report contains the bulk of our documentation, please be sure to also review the following files for complete documentation:

- README.md
- criteria-checklist.pdf
- site-map.pdf

The layout document can be accessed at the [cosc 360 project figma](#). This figma file contains the up-to-date planned layout design of the website. This includes elements, sizes and placement. While accessing the file you are able to enter presentation mode to play around with the design as it would be as a real web application. The site map can also be accessed from this link.

To get the most up-to-date code and documentation, please check out our GitHub repo:
<https://github.com/JadenBalogh/cosc360-project>

Project Proposal

For this project, our group decided to take on the first option of the given project, the discussion forum website. We have created *Postn* to allow registered users to participate in online discussions in a similar format to Reddit. Unregistered users will be allowed to view the discussions but will be unable to comment or create their own posts. Users can register, create discussions (image or text-based), and reply to posts/comments. *Postn* will support multiple users. *Postn* will also be updated in real-time using an asynchronous approach.

Requirement List

Non-User:

- View all discussion posts.
- Search for discussions.
- Create an account.
- Can navigate to any part of the website regardless of the current page.
- Sort feed by activity.
- Advanced styling.
- Collapsible threads.
- Resolve user problems.

User:

- View all discussion posts.
- Create their own posts (with images and text).
- Comment on posts.

- Reply to comments to create a thread.
- Update their profile (username, email, password, profile picture).
- Can navigate to any part of the website regardless of the current page.
- Must maintain state between page navigation.
- Search for discussions.
- Simple login with email and password.
- Password recovery if required.
- Sort feed by activity.
- Advanced styling.
- Collapsible threads.
- Resolve user problems.

Admin:

- Search for a user by name, email, or post.
- Enable/disable users.
- Edit/remove posts and comments.

Technical Implementation

Tech Stack

Postn boasts a sleek, modern design built on React and styled with Tailwind CSS. Our front-end also uses Headless UI to supplement some of our core components, as well as Heroicons to ensure consistent icon design.

On the back-end, we are running a Node.js server powered by Express. This connects the user to our Postgres database using an ORM called Sequelize for simple and streamlined data management.

Code Structure

Our codebase is divided into two distinct parts: the **client** and the **server**.

Client

The client manages all the code to make the site dynamic and responsive. There are several components to this: navigation with React Router, AJAX-style rendering with React hooks, and RESTful API calls to the server.

When a user accesses the site, React Router decides which component to render based on the current URL. For finer control over navigation options, we use the `useHistory` hook to implement custom page redirection and undo options.

Once a user has been routed to the correct page, state and effect hooks are used to make asynchronous API calls to our server and update the state of the user interface without impeding the user's actions. All of our pages - such as the feed, profile or post view - take advantage of this system to both reduce wait times and ensure that the client always has the most up-to-date data from the server.

Server

The server manages the connection to our database and allows the user to access and modify data easily through predefined routes. The main `server.js` file routes all client requests to their appropriate views, which in turn communicate with the necessary database models defined with Sequelize. This hierarchical design reduces dependencies between our data and our API, making changes less risky and improving re-usability between layers.

We also use several middlewares in our Express server to improve security, namely CORS and JWT authentication. CORS gives us control over who can connect to the API and ensures that external sources cannot send the server fake data from outside our client. JWT authentication is used on certain routes - such as editing one's profile - to validate the current user with a hashed authentication token that is assigned on login and passed between the client and server on each request.

Site Hosting

Postn is hosted with Heroku at <https://postn-dev.herokuapp.com/>. This includes both our server as well as our postgres database for general access from anywhere.

In production, the site functions slightly differently than a locally hosted deployment: rather than having a separate client and server running on separate ports, the production environment serves the client as a static bundle. This means that the user is actually connecting directly to the server-side, which sends the browser the root client file (`index.js`), leaving React Router to interpret the given URL and render the correct component. This production environment is implemented using Express to serve a static build for any path received by the server.

Server API

This is a summary of *Postn*'s server-side API and current functionality.

Routes

These are the public API paths which the client can use to receive information from the server.

Account

GET

/accounts/profile - Authenticate the user and return their profile information

POST

/accounts/login - Login and authenticate the user with a JWT token

/accounts/signup - Create a new user and save their information to the database

/accounts/password-recovery - Find a user's account information in the database and send them a password recovery email

/accounts/activate - Admin-only. Set the target user account as active

/accounts/deactivate - Admin-only. Set the target user account as inactive

PUT

/accounts/profile - Authenticate the user and update their profile information

Posts

GET

/feed/get-feed - Return all posts with optional filtering by search text and ordering by date

/feed/get-post - Retrieve a post by id from the database and return its data

/feed/comments - Return all comments for a given post id

POST

/feed/publish-post - Authenticate the user and create a new post

/feed/add-comment - Authenticate the user and create a new comment on the given post

PUT

/feed/edit-post - Authenticate the user and update the given post

/feed/edit-comment - Authenticate the user and update the given comment

DELETE

/feed/delete-post - Authenticate the user and delete the given post

/feed/delete-comment - Authenticate the user and delete the given comment

Database

These are the database models accessible through the server API.

Models

User - { email, password, name, image, isAdmin, isActive } → Posts, Comments

Post - { title, image, link, body } → User, Comments

Comment - { text } → User, Comments

Summary of features implemented

Non-User:

- View all discussion posts.
 - Posts are visible on the main page.
- Search for discussions.
 - Using the search bar at the top of the main page.
- Create an account.
 - Accessible through the create account button on the top-right of the main page.
- Can navigate to any part of the website regardless of the current page.
 - Navbar on the top exists on all the pages. Users can navigate to every page they have access to.
- Sort feed by activity.
 - Feed on the main page is sorted by activity.
- Advanced styling.
 - All the pages have a modern consistent styling.
- Collapsible threads.
 - All the threads are collapsible.
- Resolve user problems.
 - Users can ask to recover their password through email.

User:

- Create their own posts (with images and text).
 - Logged in users can create their posts that can possibly have images.
- Comment on posts.
 - Users can comment on all the posts.
- Reply to comments to create a thread.
 - Users can reply to comments to create a thread with no limit on the depth of the chain.
- Update their profile (username, email, password, profile picture).
 - Users can update all the information on their profile at will.
- Must maintain state between page navigation.
 - Sessions are stored to maintain the state between page navigation including keeping the user logged in.
- Search for discussions.
 - Using the search bar at the top of the main page.
- Simple login with email and password.
 - Users can log in with email and password.

Admin:

- Search for a user by name, email, or post.
 - Admins have an extra tab on the navbar to access the admin page where they can see users and search through them by name, email, or post.
- Enable/disable users.
 - Admins can enable/disable users in the admin page
- Edit/remove posts and comments.
 - While users can always delete/edit their own posts, admins can delete/edit all the posts.

Walkthrough

1. This is the postn homepage:

postn.

Search Login Register

Post 1
By [@second](#)
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vulputate eget velit vitae congue. Quisque iaculis tortor elit, vitae tincidunt libero condimentum in. Morbi sapien purus, accumsan id turpis et, rutrum imperdiet lorem. Etiam hendrerit ligula sed risus blandit, in efficitur nisi venenatis. Mauris rutrum posuere aliquam. Ut vehicula, metus et viverra ultrices, elit odio interdum ex, vitae tincidunt tellus diam sed lacus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean interdum sem sed purus malesuada, in venenatis eros tempus. Quisque gravid...

Post 1
By [@firstee](#)
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vulputate eget velit vitae congue. Quisque iaculis tortor elit, vitae tincidunt libero condimentum in. Morbi sapien purus, accumsan id turpis et, rutrum imperdiet lorem. Etiam hendrerit ligula sed risus blandit, in efficitur nisi venenatis. Mauris rutrum posuere aliquam. Ut vehicula, metus et viverra ultrices, elit odio interdum ex, vitae tincidunt tellus diam sed lacus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean interdum sem sed purus malesuada, in venenatis eros tempus. Quisque gravid...

My First Post
By [@firstee](#)
This is my first post! Yoooooooo! And I can edit it!!!!

Alert Test
By [@human1](#)
This is an alert test.

Human Post
By [@human1](#)
Hello humans, it is I, yet again. I am here to enjoy my humanness, with my

2. Clicking on a post will take you to that post's thread:

postn.

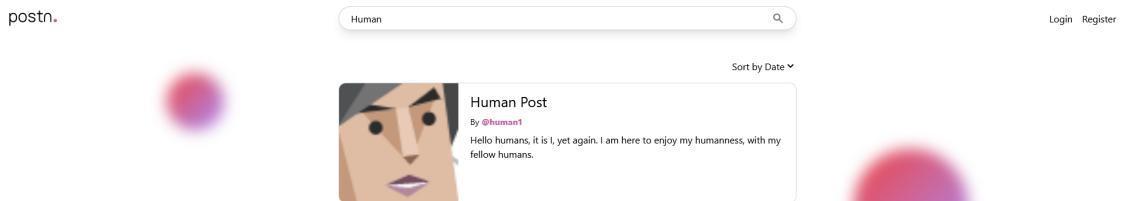
Search Login Register

Human Post
By [@human1](#)
Hello humans, it is I, yet again. I am here to enjoy my humanness, with my fellow humans.

Reply
@human1 2s
Any humans in the chat?

Reply
@firstee 2s
What's upppppp

3. You can use the search bar to find specific posts based on keywords:

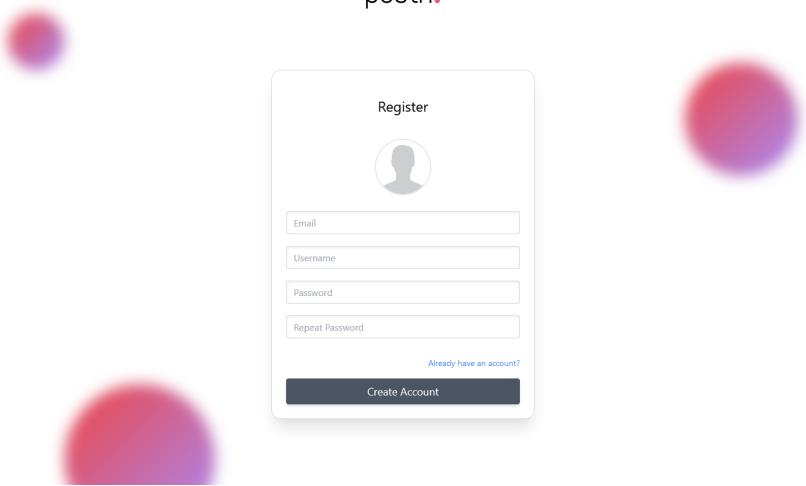


4. Clicking the login button will take you to the login page

The login page has a central 'Login' title. It contains two input fields: 'Email' and 'Password'. Below these fields are two small links: 'Forgot Password?' and 'Register'. At the bottom is a large 'Sign In' button.

5. Clicking on the register button will take you to the register section:

postn.



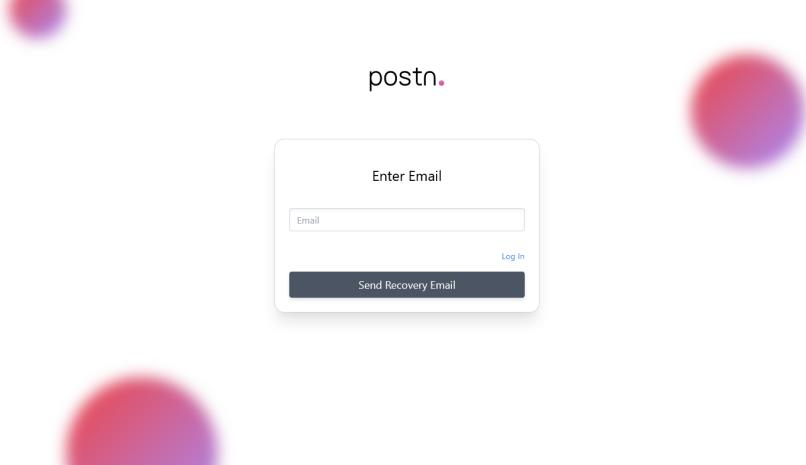
Register

Already have an account?

[Create Account](#)

6. If you forgot your email you can have it sent to you through the email recovery page:

postn.

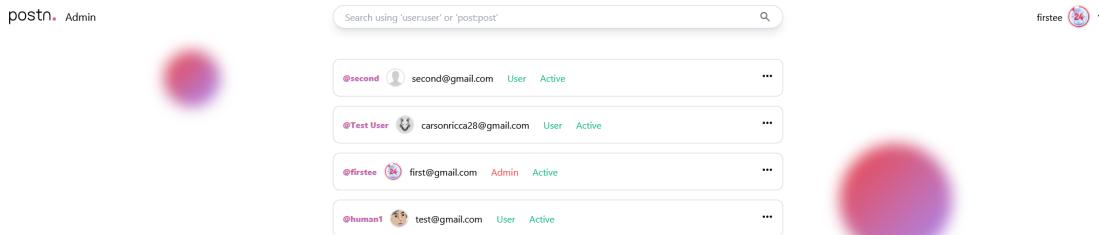


Enter Email

Log In

[Send Recovery Email](#)

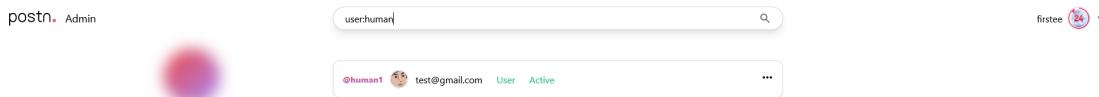
7. As an admin you can search up users to either activate/deactivate or delete their accounts:



A screenshot of a user management interface. At the top left, it says "postn Admin". In the top right, there is a dropdown menu with the name "firsee" and a red notification badge with the number "26". Below this is a search bar with the placeholder "Search using 'useruser' or 'postpost'". The main area displays a list of users in a table format:

User	Username	Email	Type	Status	Actions
@second	second	second@gmail.com	User	Active	...
@Test User	carsonricca28	carsonricca28@gmail.com	User	Active	...
@firsee	firsee	firsee@gmail.com	Admin	Active	...
@human1	human1	test@gmail.com	User	Active	...

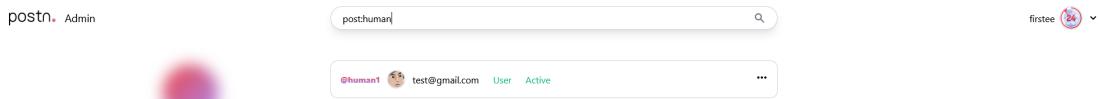
8. You can search using the user keyword:



A screenshot of a user search interface. At the top left, it says "postn Admin". In the top right, there is a dropdown menu with the name "firsee" and a red notification badge with the number "26". Below this is a search bar with the placeholder "userhuman". The main area displays a list of users in a table format:

User	Username	Email	Type	Status	Actions
@human1	human1	test@gmail.com	User	Active	...

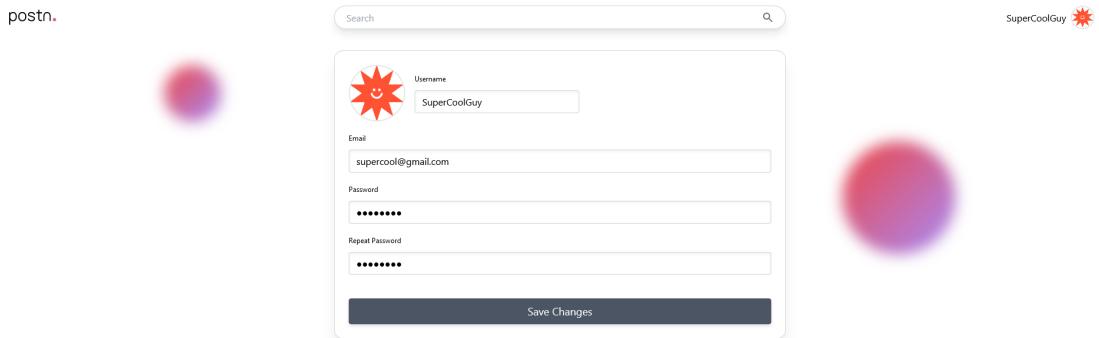
9. Or you can also search using the post keyword:



A screenshot of a user search interface. At the top left, it says "postn Admin". In the top right, there is a dropdown menu with the name "firsee" and a red notification badge with the number "26". Below this is a search bar with the placeholder "posthuman". The main area displays a list of users in a table format:

User	Username	Email	Type	Status	Actions
@human1	human1	test@gmail.com	User	Active	...

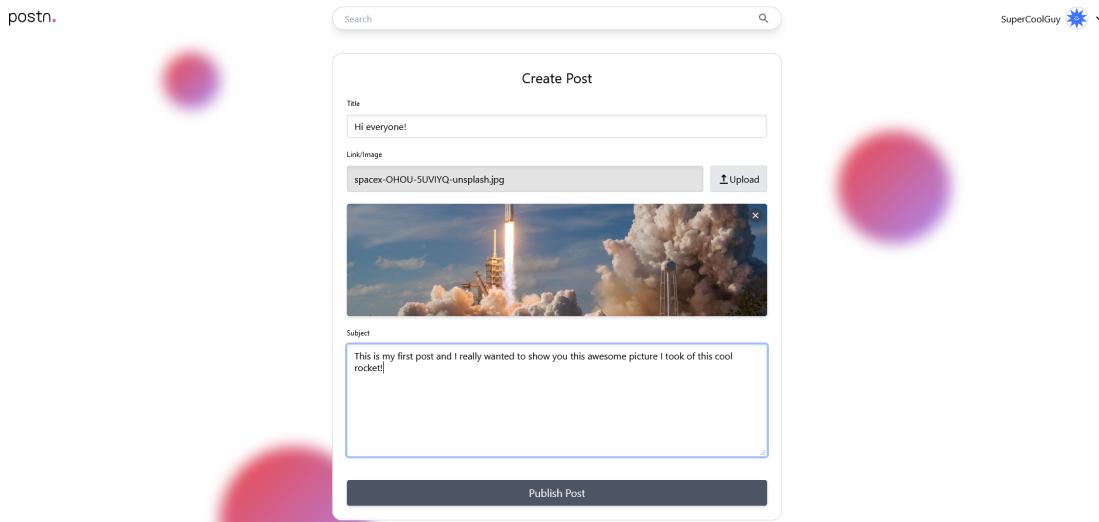
10. While logged in you can click the dropdown in the top right to access your profile details. Here you can edit the details relating to your account.



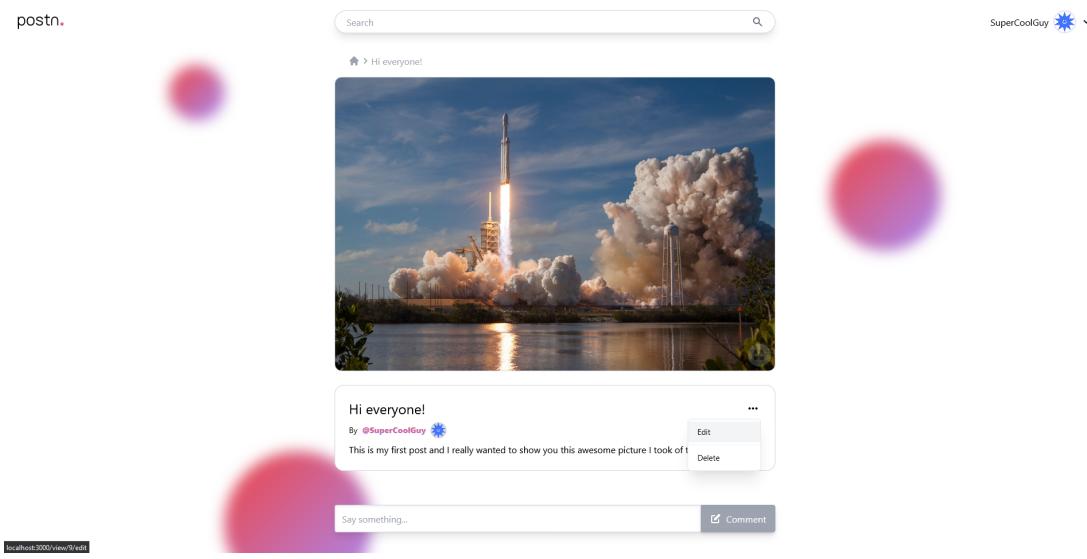
A screenshot of a user profile edit interface. At the top left, it says "postn". In the top right, there is a dropdown menu with the name "SuperCoolGuy" and a red notification badge with the number "26". Below this is a search bar with the placeholder "Search". The main area shows a form for editing profile details:

	Username	SuperCoolGuy
Email	supercool@gmail.com	
Password	*****	
Repeat Password	*****	
<input type="button" value="Save Changes"/>		

11. When logged in, by clicking the plus icon on the homepage you can create a new post:

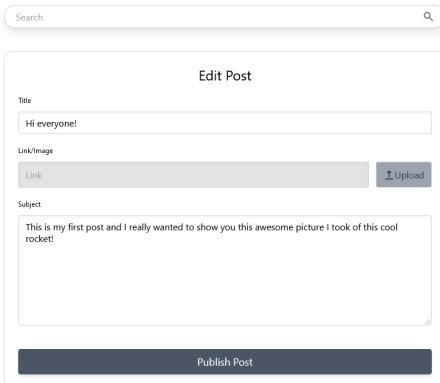


12. Once created it will take you to view that post. Then you can also click on a dropdown menu to edit or delete posts that belong to you:



13. Editing a post will allow you to change the title and the subject of the post:

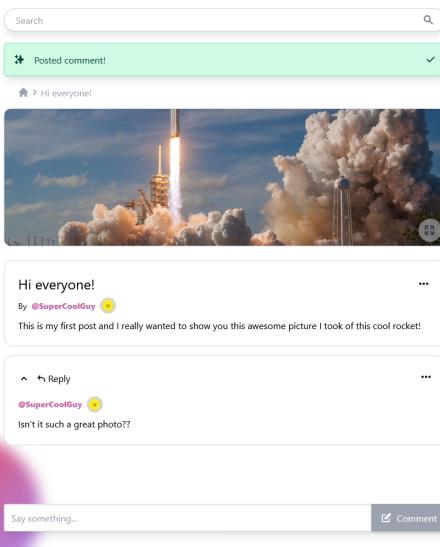
postn.



SuperCoolGuy 🌟

14. When logged in you can comment on posts to start a thread. The alert system will tell you that you have successfully posted. This alert system is built into multiple pages to let you know if something went wrong or something went right:

postn.



SuperCoolGuy 🌟

15. Homepage view when logged in with the newly created post:

postn. Search Sort by Date ^ SuperCoolGuy ...

Hi everyone!
By @SuperCoolGuy
This is my first post and I really wanted to show you this awesome picture I took of this cool rocket!

Human Post
By @human1
Hello humans, it is I, yet again. I am here to enjoy my humanness, with my fellow humans.

Alert Test
By @human1
This is an alert test.

My First Post
By @firstee
This is my first post! Yoooooooo! And I can edit it!!!!

Post 4
By @second

16. When a new post has arrived the alert system will tell you and you can click it to load the new posts:

postn. Search Sort by Date ^ SuperCoolGuy ...

Hi everyone!
By @SuperCoolGuy
This is my first post and I really wanted to show you this awesome picture I took of this cool rocket!

Human Post
By @human1
Hello humans, it is I, yet again. I am here to enjoy my humanness, with my fellow humans.

Alert Test
By @human1
This is an alert test.

My First Post
By @firstee
This is my first post! Yoooooooo! And I can edit it!!!!

Post 4
By @second