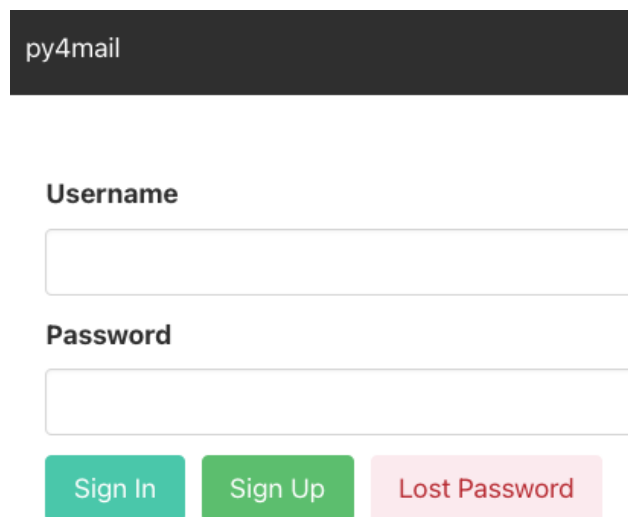# py4mail

Description:

py4mail is a locally-based email application specifically developed for facilitating communication between users. Inspired by popular email services like Gmail, Py4Mail offers a range of standard features including support for multiple user accounts, seamless sending and receiving of emails, as well as the ability to block and unblock users. The application is built using a combination of Vue.js, a JavaScript framework for building user interfaces, and py4web, a Python framework. This documentation provides a comprehensive guide on how to effectively use and navigate Py4Mail's functionalities.

Functionality:

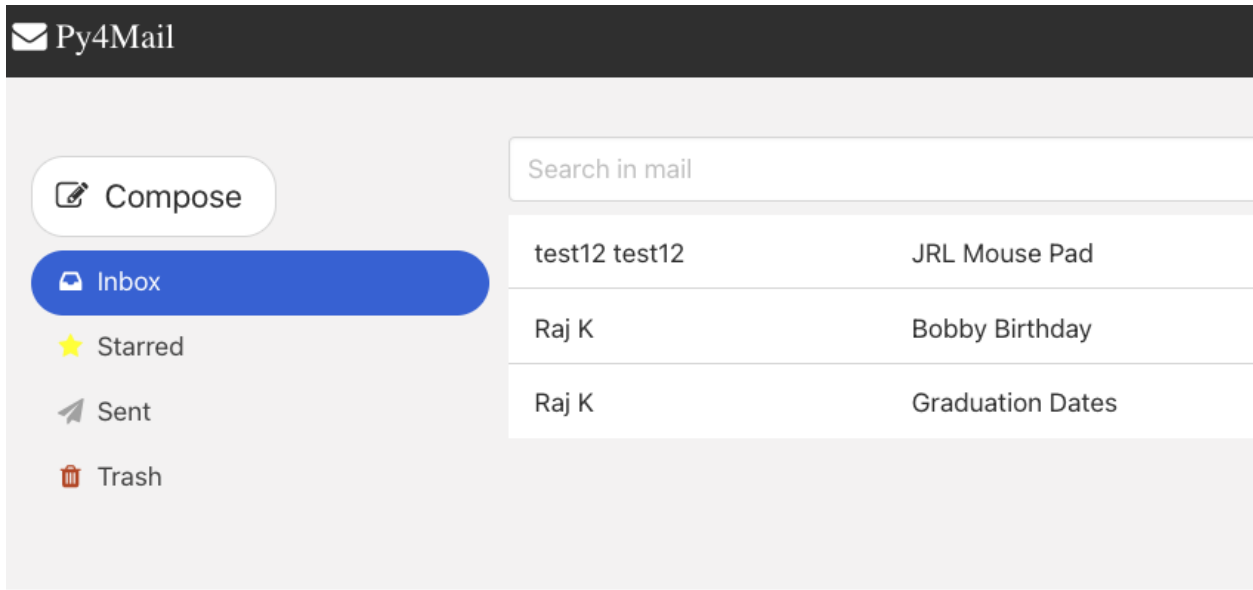Py4Mail initiates with a login page where users can securely access the application.

For the context of understanding how to navigate through the application, let's use an existing user.
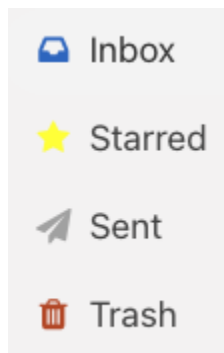
Username: test
Password: test

When login in we can see the contents of the email application.

To view other content such as Starred, Sent, and Trash press the buttons.

To create a message. Press the Compose button. Enter the username of a user that exists in the database. In this case, create another user, and send an email to that user.

By selecting a specific email, you can see the content and body of the email.

Graduation Dates

Raj K
test test

When is Graduation?

Based on the contents of the email, you can reply, forward, block, star, or trash the email.

↩   ★   ⊘   🗑   ➦   2023-06-09 16:13:27

This introduction should get you to be well-versed in the basics of py4mail.
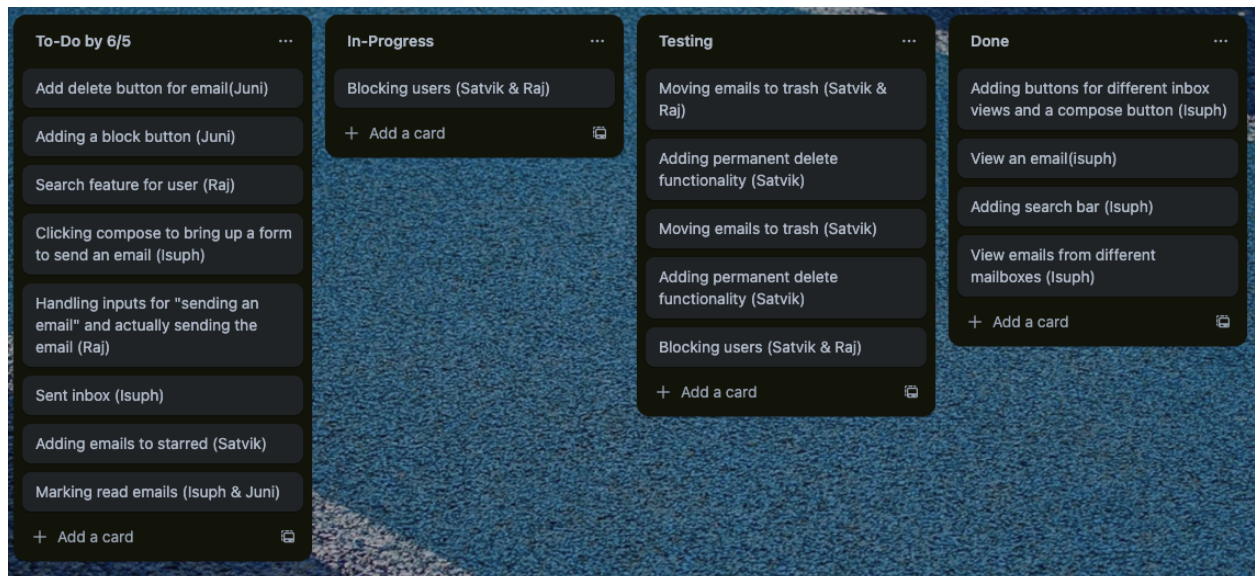
Design of py4mail

Design:

During the design and implementation process of the email web application, the team collaborated to ensure a user-friendly and functional experience.
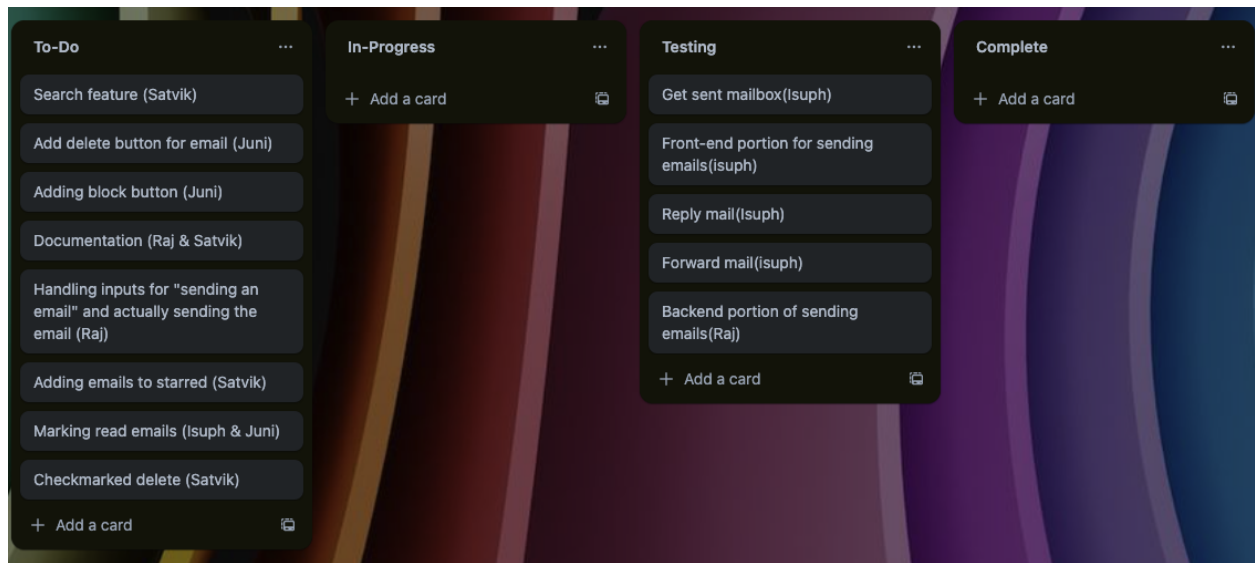- Design: The team conducted brainstorming sessions to gather ideas and insights on the key features and functionality of the email web application. We review our favorite email applications and used ideas to implement for Py4Mail
- User Interface Design: The team focused on creating a user interface. The design aimed to provide a clean and organized layout, making it easy for users to navigate and understand the application's functionalities.
- Development and Implementation: Once the design was finalized, the team used the design and specifications to implement the user interface and functionality of the email web application. This ensured that the design was translated into code, taking into account responsiveness and compatibility across different devices and browsers.

To design this project, our team decided to introduce Trello Boards to give the tasks for members to implement. The following are two Sprint boards that our team used to create this project.

Sprint 1

Sprint 2

These boards served as a valuable tool to streamline our group tasks and effectively track the features that required development.

User Stories:

As a new user of the email web application, the user should be able to easily create an account. On the starting page, they can log in with existing credentials or choose to sign up for a new account by clicking the "sign up" option, which will redirect them to a sign-up page.

After successfully creating an account, the user should be directed to the inbox page. Here, they can access various functionalities. They should be able to view a list of received emails, click on individual emails to see more details and the email content on a separate page, and have the option to delete or mark emails as starred from the list.

In addition to the email list, the user should have a sidebar available with buttons for quick navigation. The buttons should include options to view sent, starred, deleted, and sent emails.

At the top of the email list, there should be a convenient search bar that allows the user to search for specific users within their inbox. This feature helps them easily locate emails, similar to other popular email applications.

When selecting a specific email from the list, the user should have the ability to take actions such as trashing, blocking, or starring the user. This allows them to review the email content and make decisions on where the email should be directed.

Team Dynamics:

The team maintained a consistent meeting schedule of two to three times a week, which included stand-ups, to review the project's progress and share updates. We synchronized our schedules to ensure the completion of various project components, making sure accountability and effective communication within the team. Additionally, we utilized Trello Boards to track the status of features, ensuring that all team members regularly updated their progress. This facilitated transparency and kept everyone well-informed throughout the project.