

# Design Decisions

## 1. Setup

- Followed the provided instructions to set up a React app for the frontend.
- Created a single Django app named user using Django's built-in User model to handle all user-related functionalities.
- Utilized Django Rest Framework (DRF) to create APIs for user authentication and registration.

## 2. API Design Choices

- Redirected all initial traffic to the index page, which serves the login page for user authentication.
- Configured **GET** requests to be handled by the React frontend, enabling the user to access login and registration pages directly.
- Designed the user model to handle only **POST** requests for user authentication and registration.
- Implemented **field validation** (username, password, email) in the user model, returning a bad request response for invalid data.
- Set up the logout functionality to redirect users back to the login page.

## 3. User Experience / User Interface Design

- Applied a blue and white color scheme to the application to align with Hosstinn's branding style.
- Included a registration link on the login page for users who do not have an account.
- Implemented a feature to redirect already logged-in users to a designated page when they attempt to access the login or registration pages again.

## 4. Database

- Utilized Django's default **SQLite** database for data storage.
- Added a **script** to clean up the database as needed.  
(cleanup\_db.sh)