

---

# API Documentation

## Base URL

http://127.0.0.1:5000/

---

## Routes

### 1. / (Login Page)

- **Method:** GET
  - **Description:** Returns for users and admins the login page.
  - **Response:** Returns the `index.html` template.
- 

### 2. /indexuser (User Dashboard)

- **Method:** GET
  - **Description:** Shows the client dashboard in case logged in as a client. Diverts to login on the off chance that not verified.
  - **Response:**
    - Success: Returns the `indexuser.html` layout.
    - Failure: Diverts to /.
- 

### 3. /indexadmin (Admin Dashboard)

- **Method:** GET
  - **Description:** Shows the admin dashboard in case logged in as an admin. Diverts to login in the event that not confirmed.
  - **Response:**
    - Success: Returns the `indexadmin.html` layout.
    - Failure: Diverts to /.
-

#### 4. `/login` (Handle Login)

- **Method:** `POST`
  - **Description:** Verifies the client or admin utilizing the e-mail and watchword.
  - **Request Parameters:**
    - `email` (string): Email address of the user/admin.
    - `pswd` (string): Password of the user/admin.
  - **Response:**
    - Success: Redirects to `/indexuser` (for users) or `/indexadmin` (for admins).
    - Failure: Returns an "Invalid login" message.
- 

#### 5. `/signup` (Handle Signup)

- **Method:** `POST`
  - **Description:** Registers a unused client or admin based on the mail sort..
  - **Request Parameters:**
    - `email` (string): mail address for signup. Must include `@admin` or `@user`.
    - `pswd` (string): Password for the account.
  - **Response:**
    - Success: Redirects to `/` (login page).
    - Failure: Returns an error message:
      - "Invalid email. Must contain `@admin` or `@user`."
      - "Email already exists. Please try again."
- 

#### 6. `/logout` (Logout)

- **Method:** `GET`
  - **Description:** Logs the user/admin out and clears the session.
  - **Response:** Redirects to `/`.
-

## Database

### Tables

#### 1. `users`

- Columns:
  - `id` (INTEGER, Primary Key, Auto Increment)
  - `email` (TEXT, Unique, Not Null)
  - `password` (TEXT, Not Null)

#### 2. `admins`

- Columns:
    - `id` (INTEGER, Primary Key, Auto Increment)
    - `email` (TEXT, Unique, Not Null)
    - `password` (TEXT, Not Null)
- 

## Session Management

- **Session Key:** `user_type`
    - Values:
      - `"admin"`: Indicates the logged-in user is an admin.
      - `"user"`: Indicates the logged-in user is a standard user.
- 

## Error Handling

- In case verification falls flat, clients get a message showing invalid qualifications.
  - In the event that e-mail as of now exists amid signup, a special imperative infringement blunder message is appeared.
- 

## Run Locally

Guarantee Python and Carafe are introduced, Spare the code to a record, e.g., `app.py`.

Run the app with: `python app.p`, Access the application at <http://127.0.0.1:5000/>.

---