**TODO application**

**Use Cases:**

1. User should be able to login
2. User should be able to logout
3. User should be able to create account
4. User should be able to Forgot password
5. User Should be able Change their Password
6. User should view the todo list
7. User should be able to create todo items
8. User Should be able to Delete Todo items
9. User should be able to mark completed todo

**API Endpoints:**

**Authentication**

1. POST /create → Create Account
2. POST /login → Login User
3. POST /logout → Logout User

**User Profile**

1. GET /me → Get User Profile
2. PUT /me → Update Profile
3. PUT /me/password → Change Password

**Todo Management**

1. GET /items → Get All Todo Items
2. POST /items → Create Todo Item
3. DELETE /items/{id} → Delete Todo Item
4. PUT /items/{id}/complete → Mark Todo as Completed

**Schema:**

User Table

| **Column** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **id** | **INT** | **PK, AUTO\_INCREMENT** | **Unique User ID** |
| **fn** | **VARCHAR(50)** | **NOT NULL** | **First Name** |
| **ln** | **VARCHAR(50)** | **NOT NULL** | **Last Name** |
| **email** | **VARCHAR(100)** | **UNIQUE, NOT NULL** | **User Email (Login ID)** |
| **password** | **VARCHAR(255)** | **NOT NULL** | **Hashed Password** |
| **photo\_url** | **VARCHAR(255)** | **NULL** | **Profile Photo Path** |
| **created\_at** | **TIMESTAMP** | **DEFAULT CURRENT\_TIMESTAMP** | **Account Creation Date** |

**Todos Table**

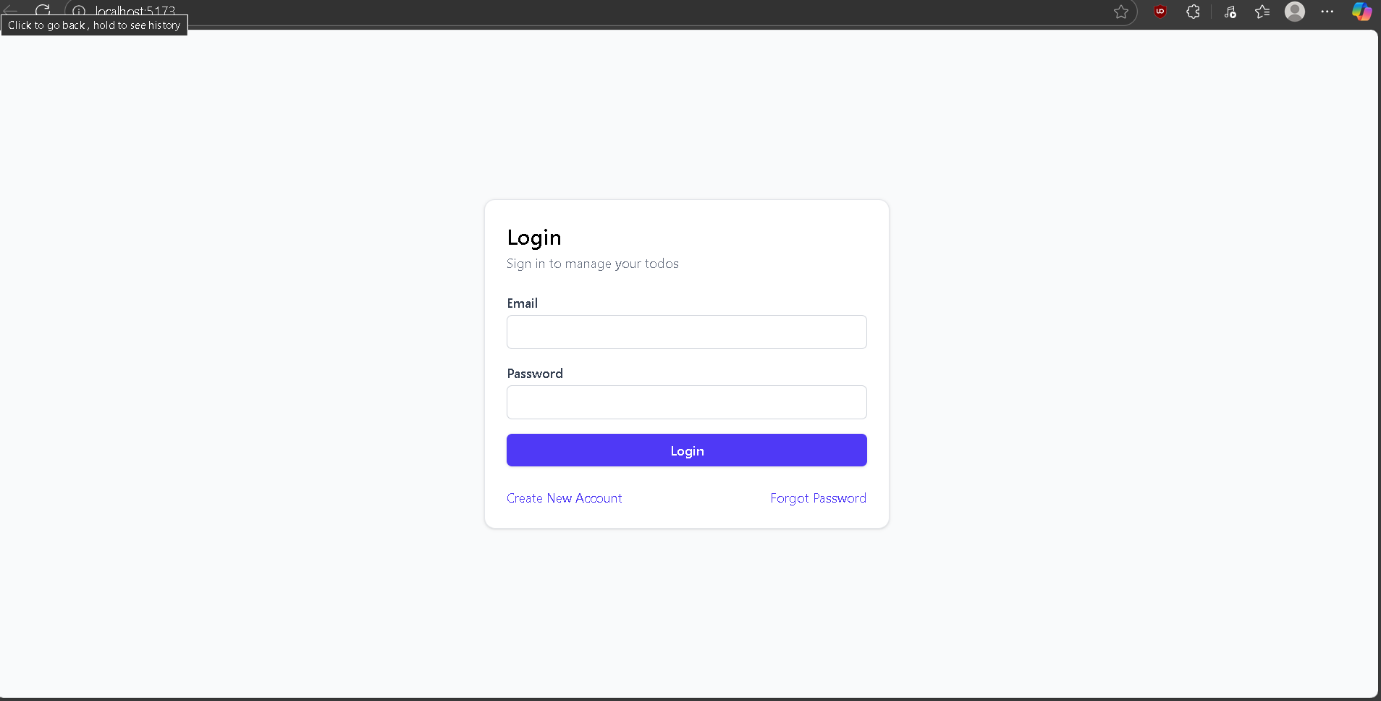
| **Column** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **id** | **INT** | **PK, AUTO\_INCREMENT** | **Unique Todo ID** |
| **user\_id** | **INT** | **FK → users(id), NOT NULL** | **User who owns the Todo** |
| **task** | **VARCHAR(255)** | **NOT NULL** | **Todo Task Description** |
| **status** | **ENUM** | **DEFAULT 'pending'** | **Task Status (pending / completed)** |
| **created\_at** | **TIMESTAMP** | **DEFAULT CURRENT\_TIMESTAMP** | **Task Creation Date** |

**Tech Stack:**

* **Frontend: ReactJS**
* **Backend: Node.js (Express.js)**
* **Database: MySQL**
* **Authentication: JWT**
* **Password Security: Bcrypt**

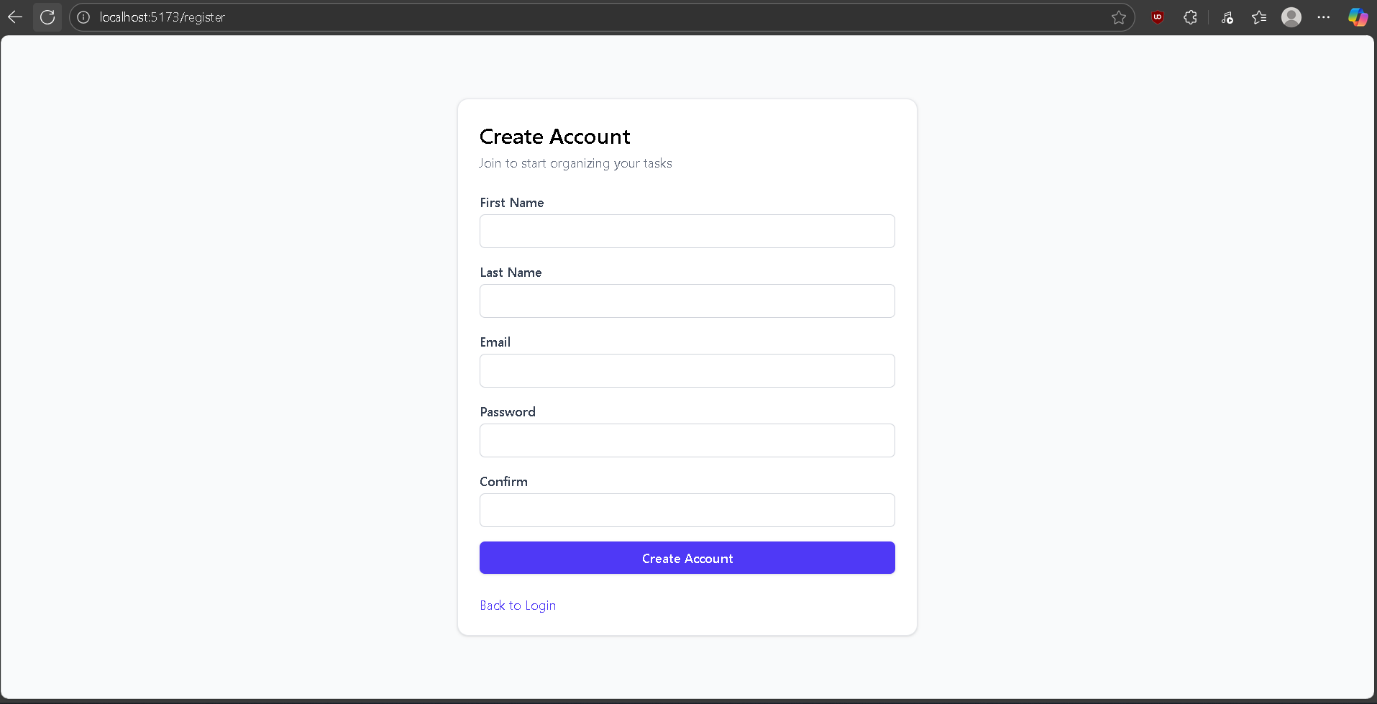
**Work Flow :**

**1. Home Page:**

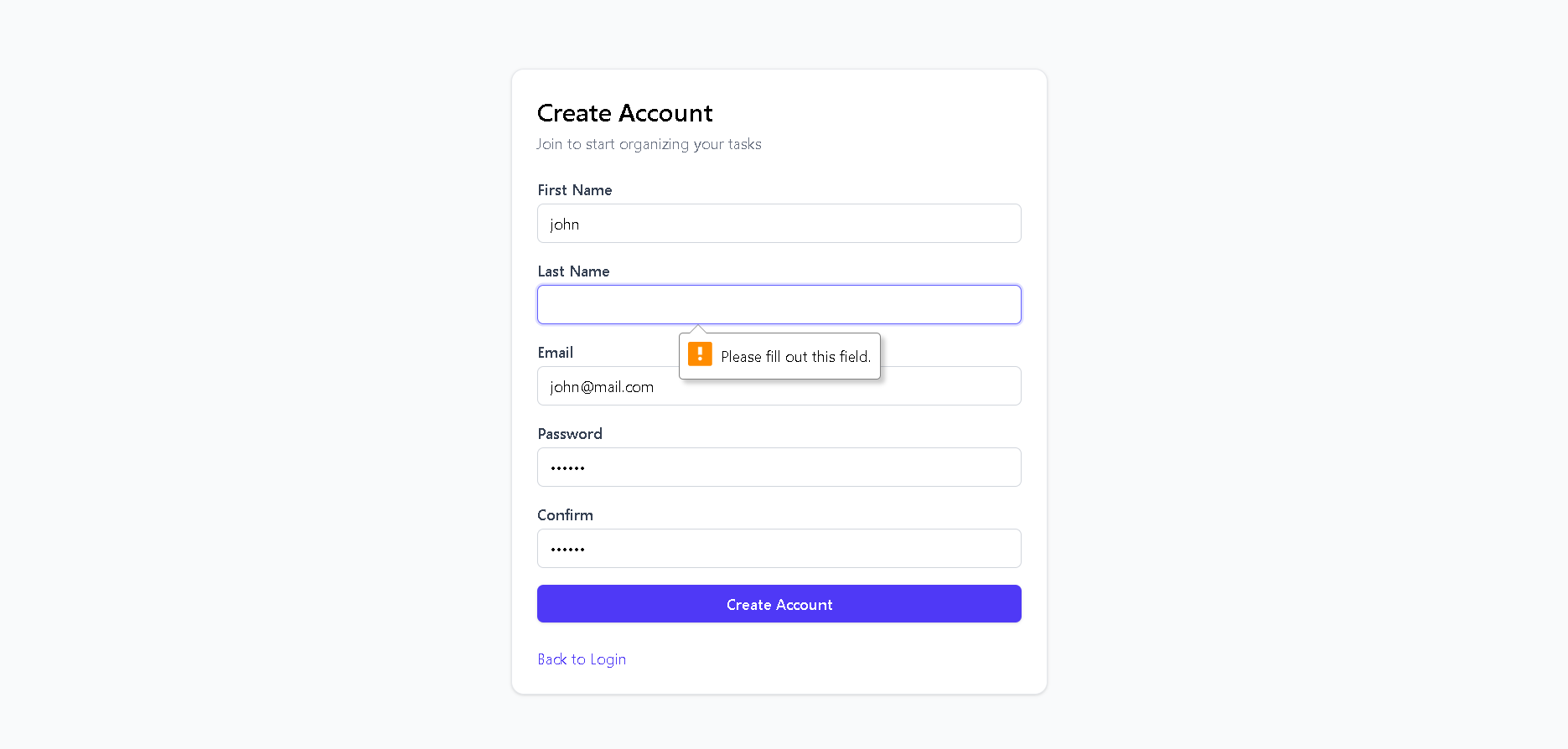
****

**2.User Registration**

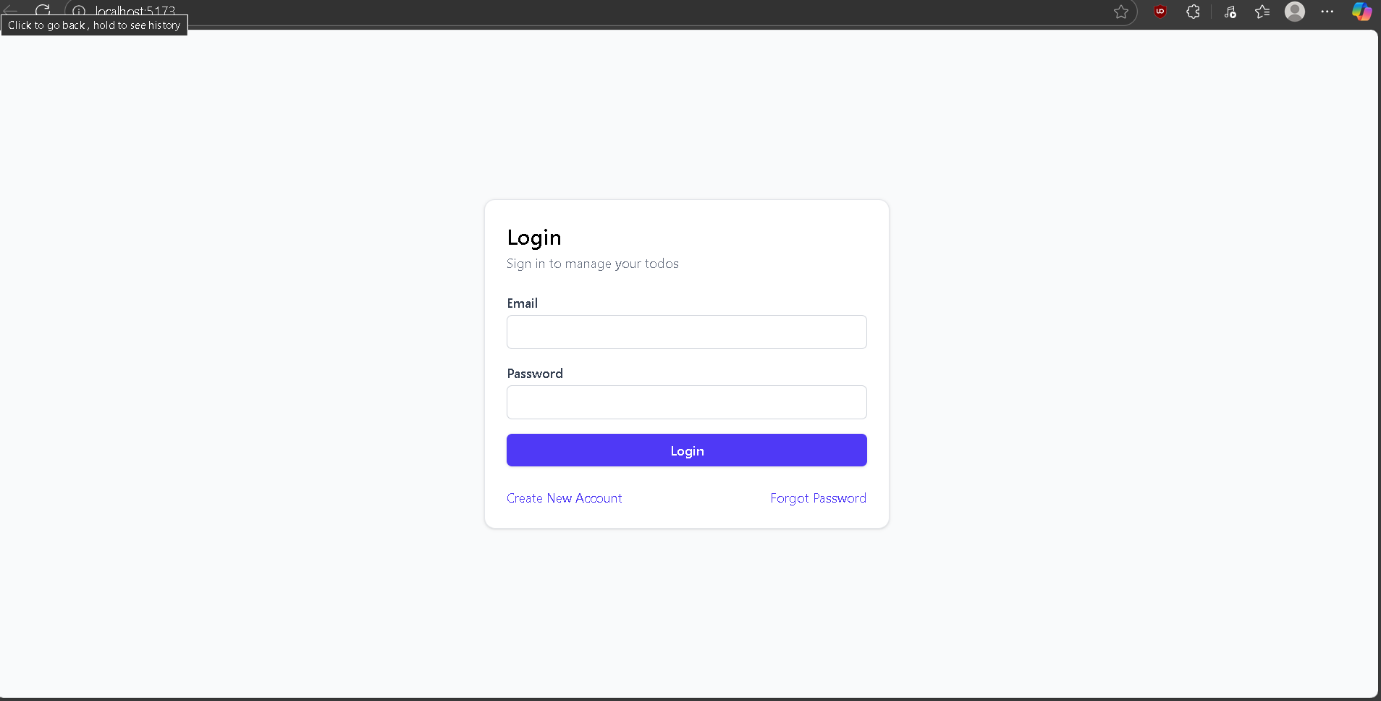
* Endpoint: POST /create
* User enters First Name, Last Name, Email, Password.
* If email is unique, account is created.
* Screenshot: Registration form.



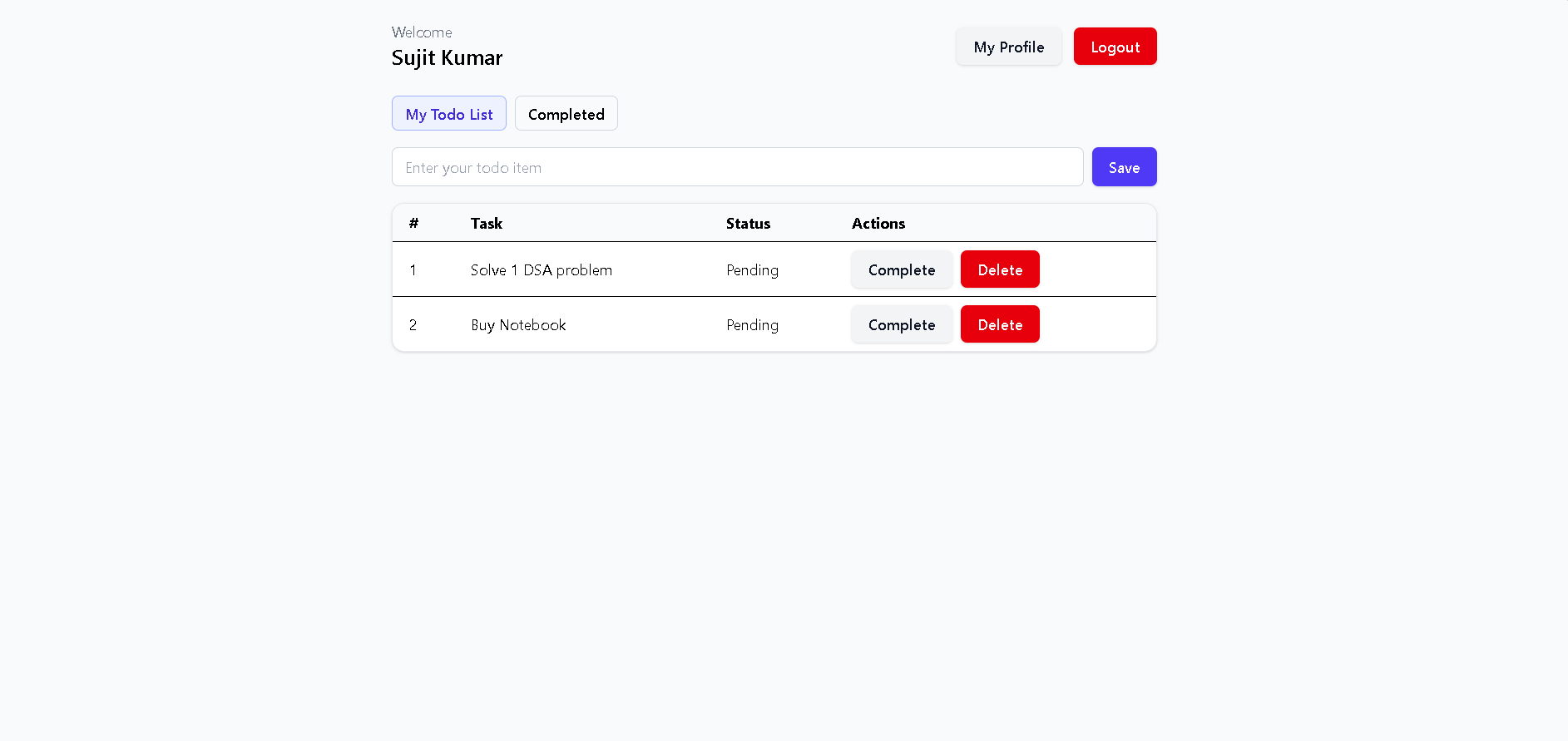
**There is validation check in every form if the input field is empty or invalid character and also for email input**



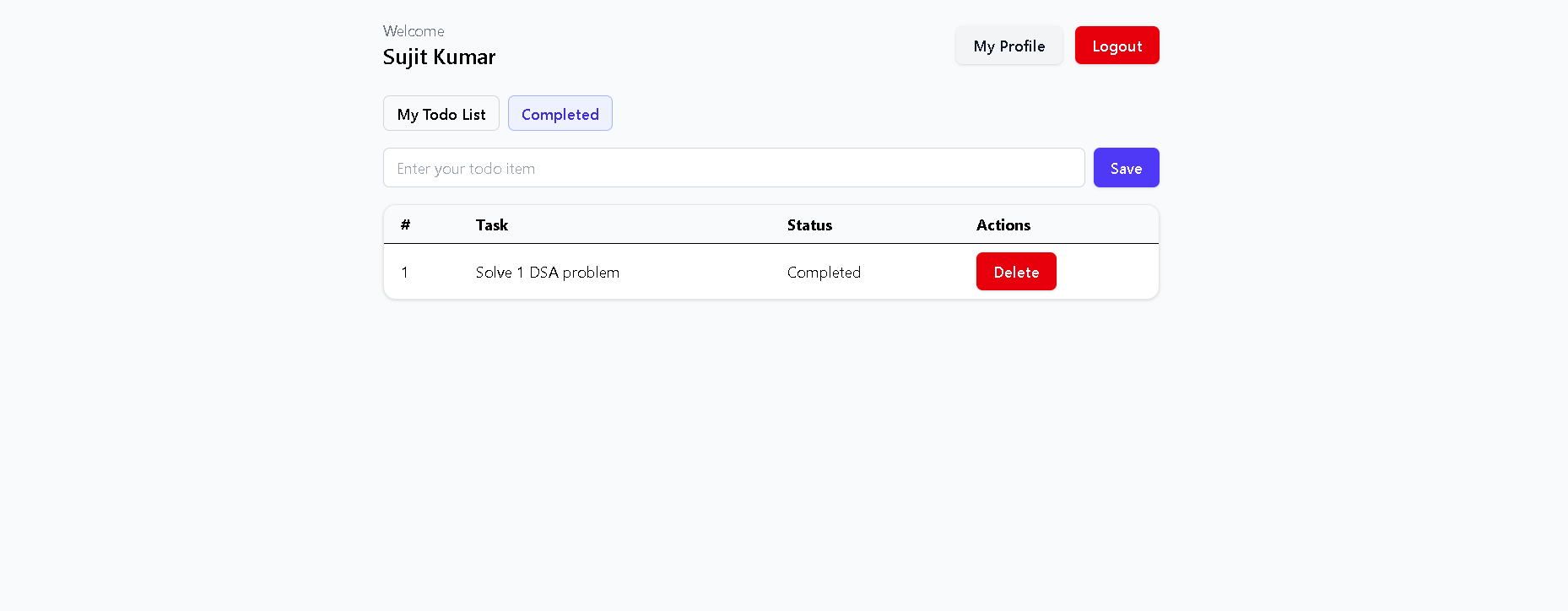
**3.User Login**

* Endpoint: POST /login
* User enters **Email + Password**.
* On success → API returns a **JWT Token**.
* **Screenshot:** Login form

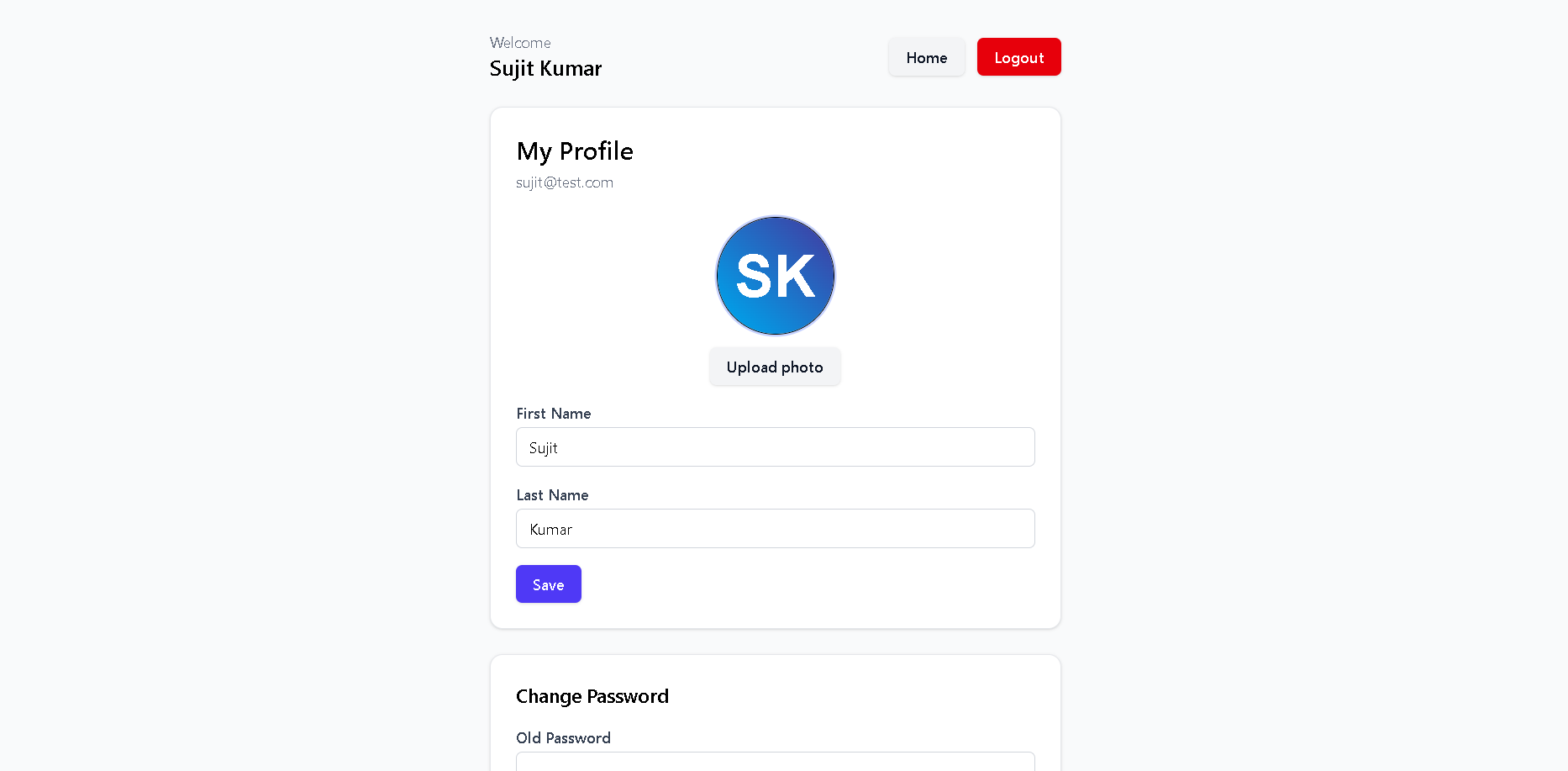
**After Login:**



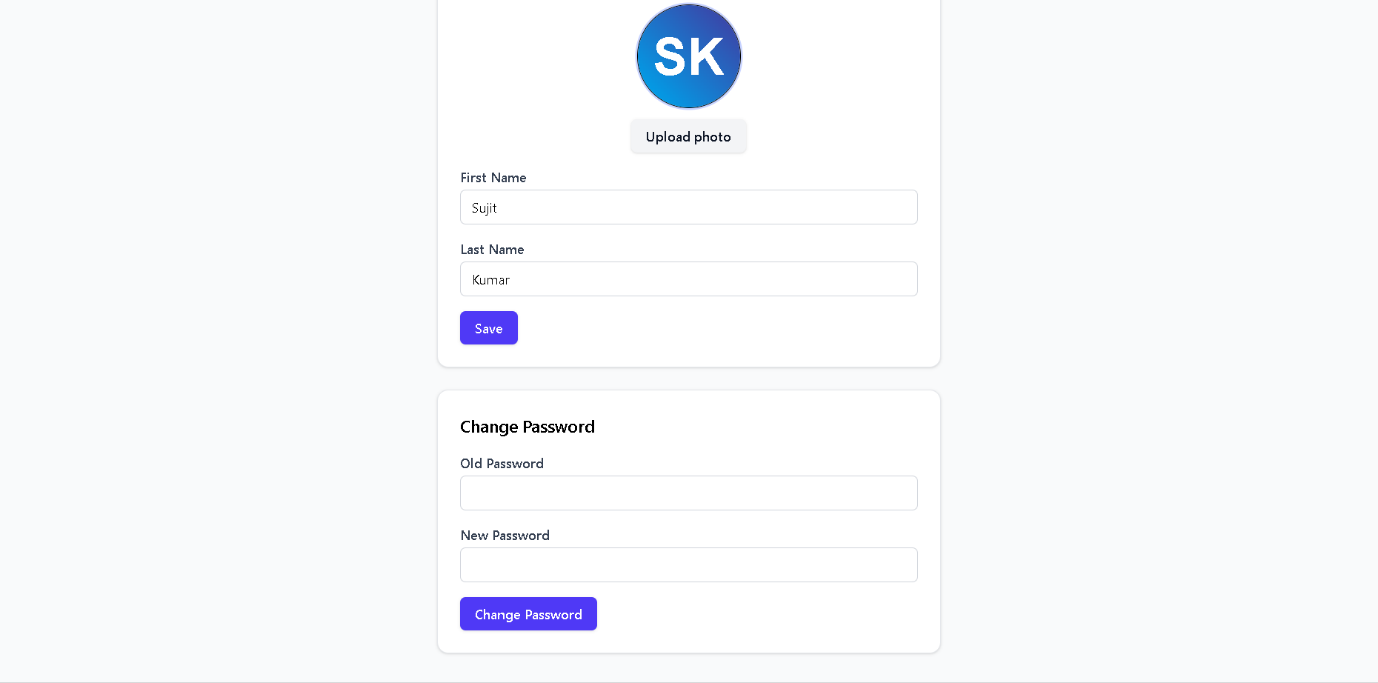
**Here User Add Todo , mark todo as Completed, delete todo, go-to their profile and user can logout.**

**When User Click on Completed the todo will move in the completed tab**

**Now In profile section user can change their name and also password**

****

**From profile page the user also change password by using old password and then new password**

****