

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

4. GET /me → Get User Profile
5. PUT /me → Update Profile
6. PUT /me/password → Change Password

Todo Management

8. GET /items → Get All Todo Items
9. POST /items → Create Todo Item

10. DELETE /items/{id} → Delete Todo Item
11. 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

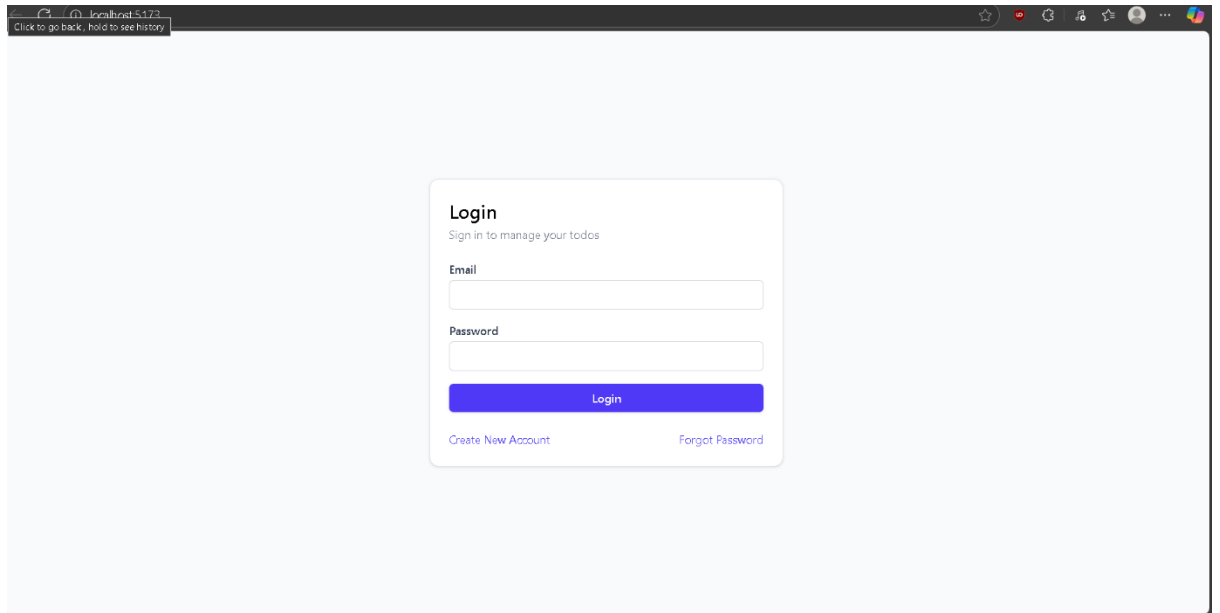
Column	Data Type	Constraints	Description
id	INT	PK, AUTO_INCREMENT	Unique Todo ID
user_id	INT	FK → users(id), NOT NULL	User who owns the Todo
task	VARCHAR(255) NOT NULL		Todo Task Description
status	ENUM	DEFAULT 'pending'	Task Status (pending / completed)
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Task Creation Date

Tech Stack:

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

Work Flow :

1. Home Page:



A screenshot of a web browser displaying a login form. The browser's address bar shows 'localhost:5173'. The login form is centered on a light gray background. It has a title 'Login' and a subtitle 'Sign in to manage your todos'. Below the subtitle are two input fields: 'Email' and 'Password'. A blue 'Login' button is positioned below the password field. At the bottom of the form, there are two links: 'Create New Account' and 'Forgot Password'.

localhost:5173
Click to go back, hold to see history

Login
Sign in to manage your todos

Email

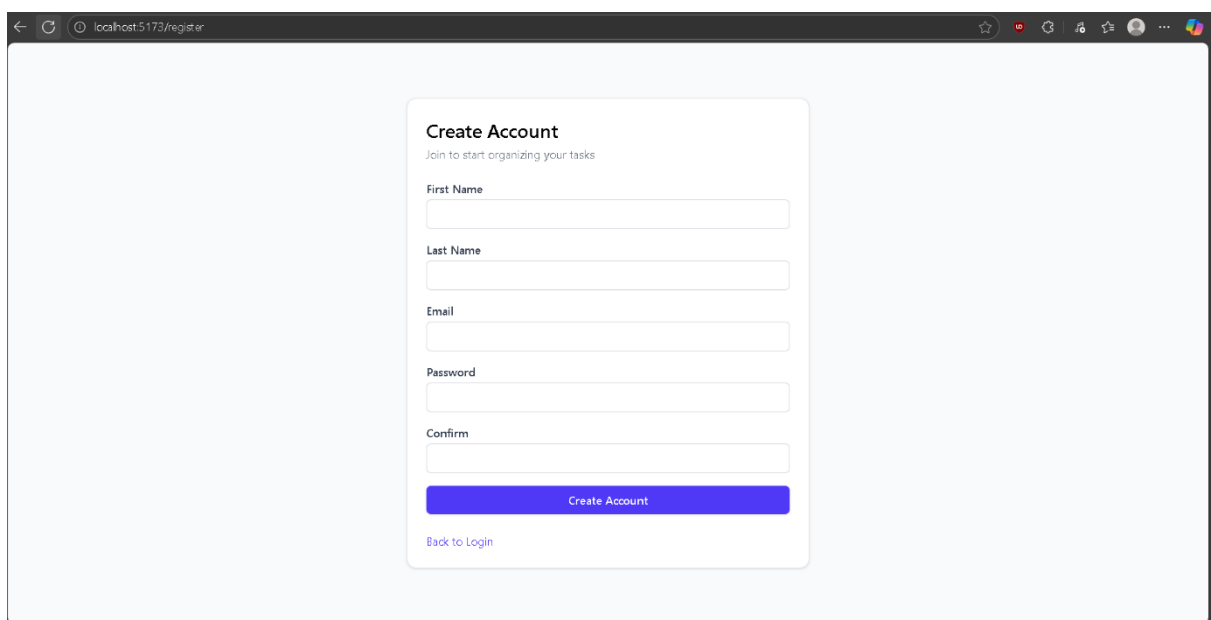
Password

Login

[Create New Account](#) [Forgot Password](#)

2. User Registration

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



A screenshot of a web browser displaying a registration form. The browser's address bar shows 'localhost:5173/register'. The registration form is centered on a light gray background. It has a title 'Create Account' and a subtitle 'Join to start organizing your tasks'. Below the subtitle are five input fields: 'First Name', 'Last Name', 'Email', 'Password', and 'Confirm'. A blue 'Create Account' button is positioned below the 'Confirm' field. At the bottom of the form, there is a link 'Back to Login'.

localhost:5173/register

Create Account
Join to start organizing your tasks

First Name

Last Name

Email

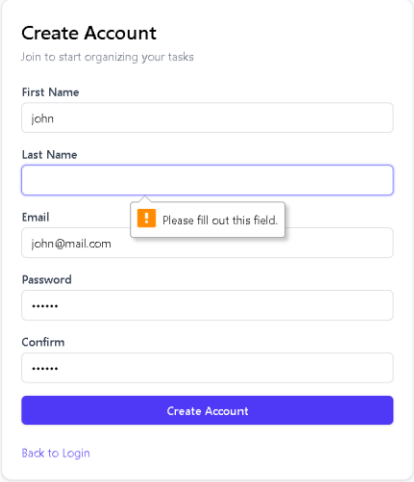
Password

Confirm

Create Account

[Back to Login](#)

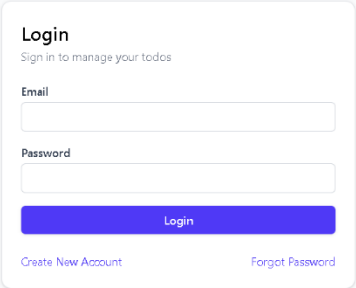
There is validation check in every form if the input field is empty or invalid character and also for email input



The screenshot shows a 'Create Account' form with the following fields: First Name (filled with 'john'), Last Name (empty), Email (filled with 'john@mail.com'), Password (filled with '*****'), and Confirm (filled with '*****'). A red error message 'Please fill out this field.' is displayed above the empty Last Name field. The form includes a 'Create Account' button and a 'Back to Login' link.

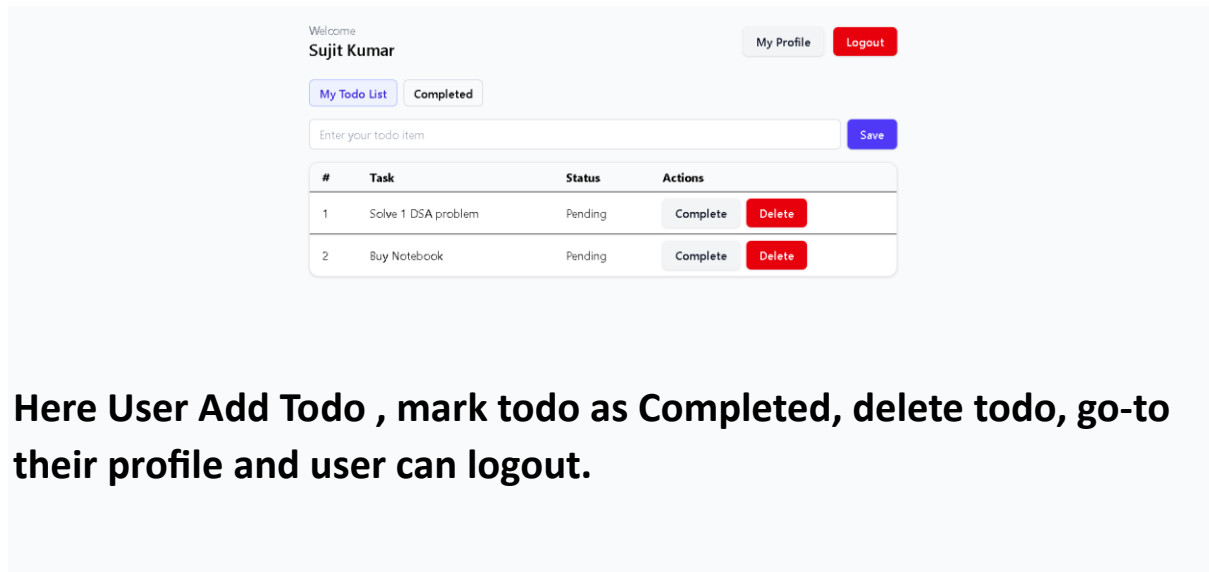
3. User Login

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

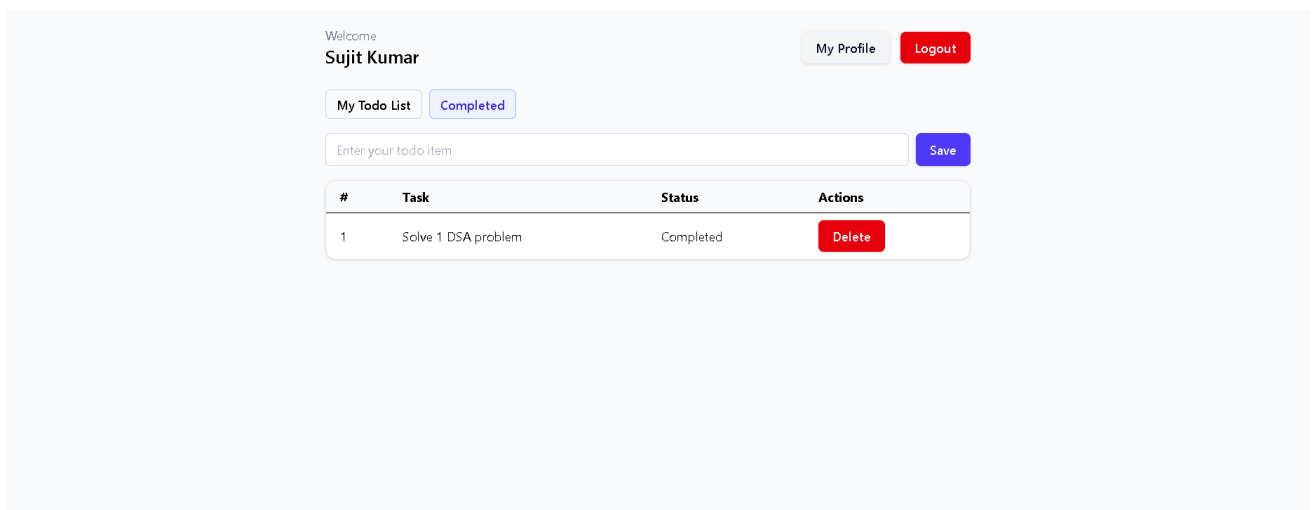


The screenshot shows a 'Login' form in a web browser window. The form has two input fields: 'Email' and 'Password'. Below the fields is a blue 'Login' button. At the bottom of the form, there are two links: 'Create New Account' and 'Forgot Password'. The browser's address bar shows 'localhost:5173'.

After Login:



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

Welcome
Sujit Kumar

HomeLogout

My Profile
sujit@test.com


Upload photo

First Name
Sujit


Last Name
Kumar

Save

Change Password

Old Password

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


Upload photo

First Name
Sujit

Last Name
Kumar

Save

Change Password

Old Password

New Password

Change Password