

# AI - HABIT TRACKER MOTIVATOR

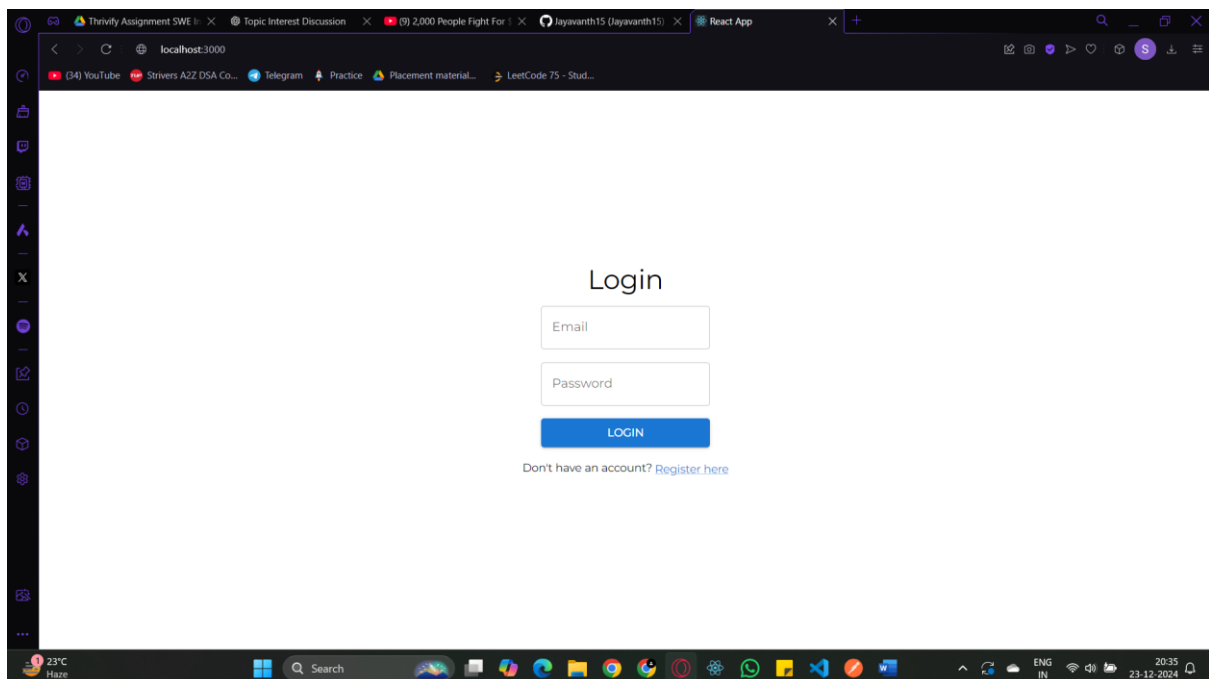
## FRONTEND

I have implemented the Login and Register pages, as well as the Dashboard page. The Dashboard consists of three components: Habits Progress, Habit List (with CRUD functionality), and AI suggestions for habits. Lastly, there's a Profile section where users can change their name and password. I've also added light mode and dark mode features.

### To access the frontend:

1. Navigate to the frontend folder: `cd frontend/habit-tracker`
2. Start the application: `npm start`

## LOGIN PAGE



# AI - HABIT TRACKER MOTIVATOR

## REGISTER PAGE:

Register

Name

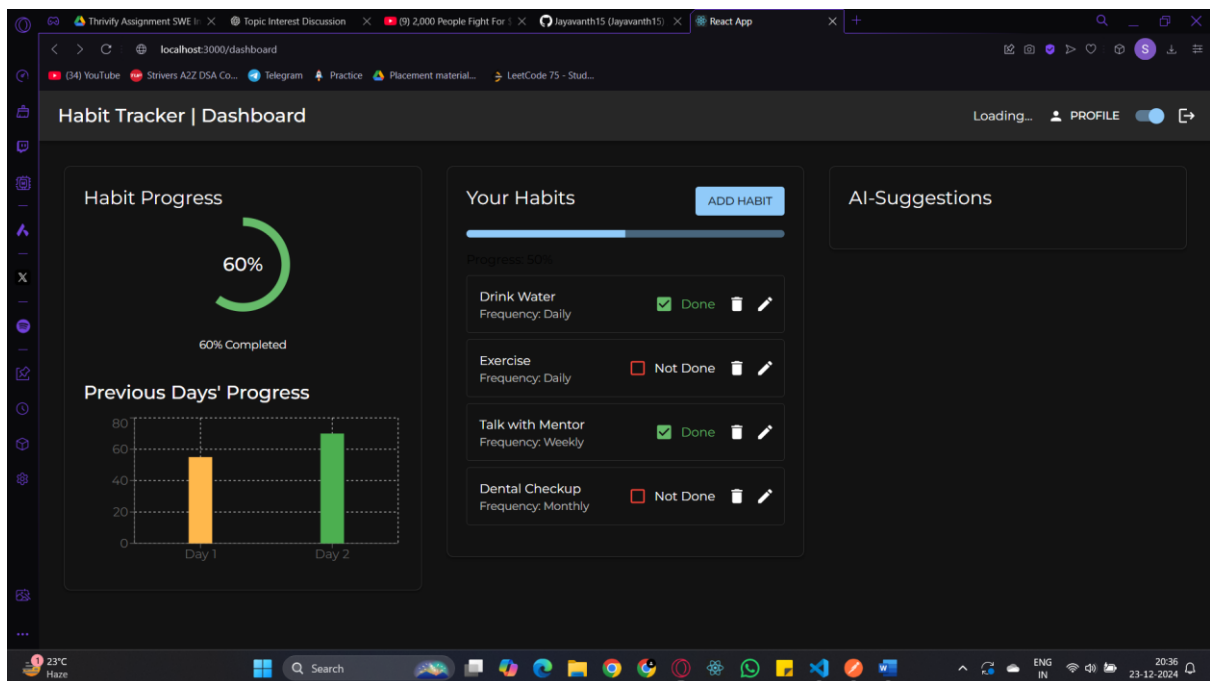
Email

Password

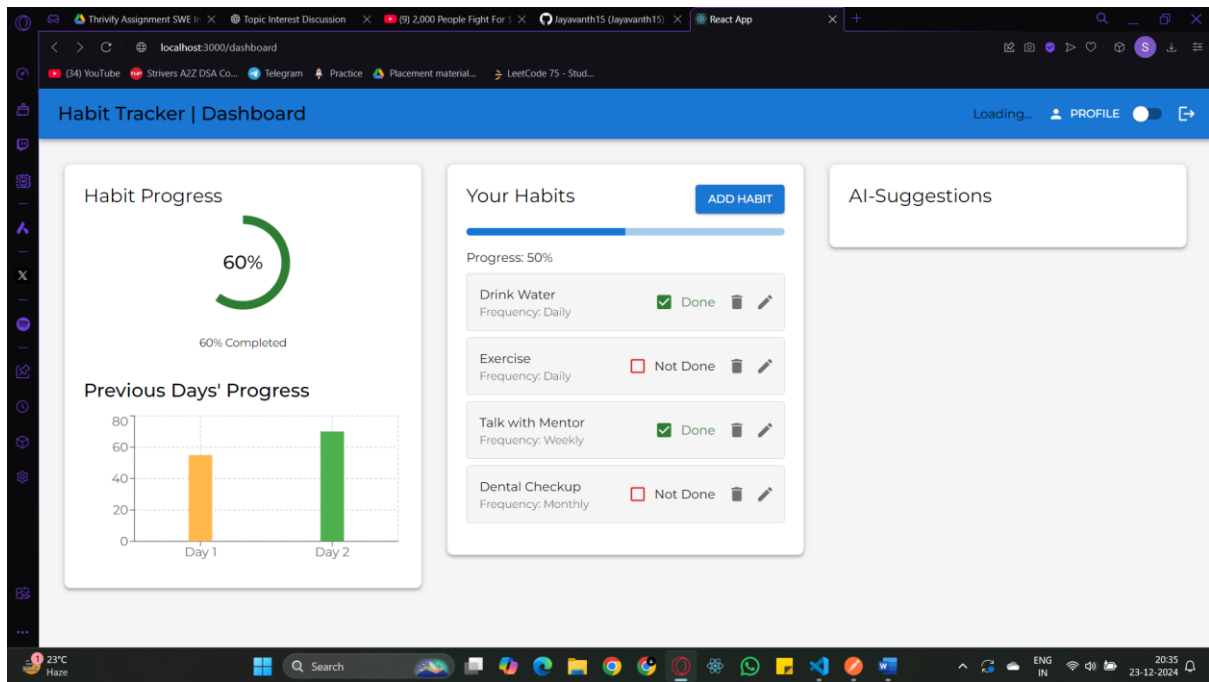
Confirm Password

REGISTER

## DASHBOARD UI:



# AI - HABIT TRACKER MOTIVATOR



## USER PROFILE PAGE:

The screenshot displays the 'User Profile' page. It features a sidebar with navigation icons, a top navigation bar with a 'Loading...' status and a 'PROFILE' link, and a main content area. The main area includes a 'Profile Details' section with a 'Previous Name:' label, a text input field for 'Name', a 'SAVE CHANGES' button, a 'Change Password' button, and an 'UPDATE PASSWORD' button. Below this section, the 'Total Progress' is displayed as 75%.

Field	Value
Previous Name:	
Name	
Change Password	
Update Password	

Total Progress: 75%

## BACKEND

In the backend, I have implemented API routes for each functionality and tested them in Postman. The APIs handle operations for users (`localhost:5000/api/users`) and habits (`localhost:5000/api/habits`). The data is stored in a local MySQL database. Additionally, I have successfully implemented user authentication in the project.

### To access the backend:

1. Navigate to the backend folder: `cd backend`

# AI - HABIT TRACKER MOTIVATOR

2. Start the server: `node app.js`

## In detail API functionality

Create a New User	<div>Method: POST</div> <div>URL: http://localhost:5000/api/users</div>	<div>{"name": "John Doe", "email": "john@example.com", "password": "password123" }</div>	<div>{ "message": "User created successfully", "userId": 12 }</div>
Login User	<div>Method: POST</div> <div>URL: http://localhost:5000/api/users/login</div>	<div>{ "email": "john@example.com", "password": "password123" }</div>	<div>{ "message": "Login successful", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJyYXVpYXQpvaG5AZXhkbXBsZS5jb20iLCJpYXQpOiJyE3MzQ5Njc5MjMlMlV4cCI6MTczNDk3MTUyM30.U4DCgYK1U_pOrBc6AuCln7DeQtkxrrP6_wqmYSGNJ2U" }</div>
Get User by ID	<div>Method: GET</div> <div>URL: http://localhost:5000/api/users/:userId</div>	-	<div>{ "name": "John Doe", "email": "john@example.com" }</div>
Update User by ID	<div>Method: PUT</div> <div>URL: http://localhost:5000/api/users/:userId</div>	<div>{ "name": "John Updated", "email": "john.updated@example.com" }</div>	<div>{ "message": "User updated successfully" }</div>
Create a New Habit	<div>Method: POST</div> <div>URL: http://localhost:5000/api/habit</div>	<div>{ "userId": 1, "habitTitle": "Morning Jog", "startDate": "2024-12-23", "frequency": "Daily", "status": "Done" }</div>	<div>{ "message": "Habit created", "habitId": 4 }</div>
Get Habits for a User	<div>Method: GET</div> <div>URL: http://localhost:5000/api/habit/:userId</div>	-	<div>[ { "habit_id": 3, "user_id": 6, "habit_title": "Drink Water",</div>

## AI - HABIT TRACKER MOTIVATOR

			<pre>"start_date": "2024-12-20T18:30:00.000Z", "frequency": "Daily", "status": "Not Done", "created_at": "2024-12-21T08:10:00.000Z" } ]</pre>
<b>Update Habit Status</b>	<b>Method:</b> GET <b>URL:</b> http://localhost:5000/api/habit/:userId	{ "status": "Not Done" }	
<b>Delete a Habit</b>	<b>Method:</b> DELETE <b>URL:</b> http://localhost:5000/api/habit/:userId	-	{ "message": "Habit deleted" }

### AI Suggesting system

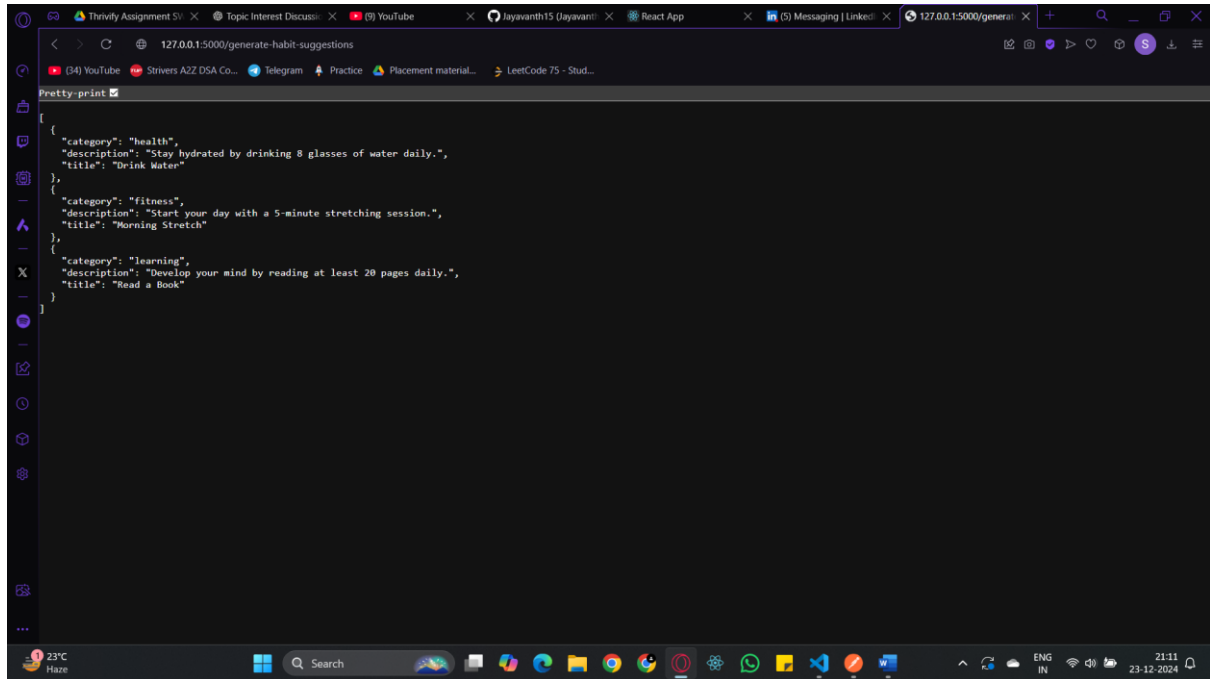
This Flask app generates habit recommendations based on a user's existing habits by comparing habit categories using cosine similarity. It loads a predefined list of habits, vectorizes their categories, and calculates the similarity between the user's habits and the available habit categories. The app then returns the top 3 habit suggestions in JSON format when the /generate-habit-suggestions route is accessed.

<http://127.0.0.1:5000/generate-habit-suggestions>

To access to Flask code:

- To start your Flask application, navigate to the folder : cd ai\_service
- Run the Flask app using: flask run

# AI - HABIT TRACKER MOTIVATOR



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/generate-habit-suggestions". The browser has several tabs open, including "Therify Assignment S...", "Topic Interest Discuss...", "YouTube", "Jayavanth15 (Jayavanth15)", "React App", "(5) Messaging | Linked...", and "127.0.0.1:5000/genera...". The main content area displays a JSON array of three habit suggestions, each with a category, description, and title. The suggestions are: 1. Health category: "Stay hydrated by drinking 8 glasses of water daily." with title "Drink Water". 2. Fitness category: "Start your day with a 5-minute stretching session." with title "Morning Stretch". 3. Learning category: "Develop your mind by reading at least 20 pages daily." with title "Read a Book". The browser's taskbar at the bottom shows the system clock as 21:11 on 23-12-2024, along with various application icons and a search bar.

```
{
  {
    "category": "health",
    "description": "Stay hydrated by drinking 8 glasses of water daily.",
    "title": "Drink Water"
  },
  {
    "category": "fitness",
    "description": "Start your day with a 5-minute stretching session.",
    "title": "Morning Stretch"
  },
  {
    "category": "learning",
    "description": "Develop your mind by reading at least 20 pages daily.",
    "title": "Read a Book"
  }
}
```