Take-Home Assignment- Thrivify AI

Role: **Full-Stack, DevOps, and AI Development Intern**
Duration: 2-3 Days
Estimated Time Commitment: 6-8 hours

📘 Objective

This assignment will assess your skills in **front-end development**, **back-end development**, and **AI/ML integration**. You'll be required to create a simple **"Habit Tracker & Motivator" Web App** where users can create and track daily habits. The platform will leverage an AI-powered suggestion system to recommend personalized habits for users.

This task will help us gauge your proficiency in **full-stack development**, **API creation**, **database management**, and **AI/ML integration**. We value clean, modular, and maintainable code that reflects your approach to problem-solving.

📋 Assignment Tasks

🛠️ The Challenge

Create a **Habit Tracker & Motivator Web App** with the following key components:

Frontend (React.js)

Backend (Node.js, Express.js, MySQL)

AI-Powered Habit Recommendation Service (Flask, Scikit-learn, Pandas, Numpy)

🎯 Key Features

Here's what we expect the web app to do:

1. Habit Tracking System

- **Create New Habit**: Users can add daily/weekly/monthly habits like "Drink Water", "Read a Book", etc.
- **Track Habits**: Users can mark habits as **done/not done** for each day.
- **View Progress**: A simple **progress bar or calendar** view that shows progress for each habit.
- **Habit Status**: Each habit should show its status (Active, Inactive, Completed).

---

2. AI-Powered Habit Suggestions

- **Personalized Suggestions**: Build an **AI-based system** to suggest new habits for users.
- AI can be **rule-based or a simple recommendation system** (like suggesting "Drink more water" if the user has health-related habits).
- Use **Flask API** to create an endpoint to get suggestions:
    - **GET /generate-habit-suggestions** — This endpoint should return a set of 3 habit recommendations.
    - Example response:

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

---

3. User Account System

- **User Registration/Login**: Implement a simple user authentication system using **JWT**.
- **User Dashboard**: Each user has a personal dashboard showing their **habit progress, suggested habits**, and the ability to add or remove habits.
- **Profile Section**: A user profile page where users can view their total progress, edit their name, and change their password.

💻 **Technical Requirements**

Frontend (React.js)

- Build a responsive **user dashboard** where users can:
    - **View habits** (habit title, progress, and status).
    - **Mark habits as completed** for the day/week/month.
    - **View suggestions** for new habits.
    - **Add/Edit/Delete habits** from their list.
    - **View progress** via charts, progress bars, or calendar views.

Backend (Node.js, Express.js, MySQL)

- **API Endpoints**:
    - **POST /api/habits** — Create a new habit.
    - **GET /api/habits** — Get a list of habits for the authenticated user.

    - **PUT /api/habits/:id** — Update the status of a habit (mark as done, edit title, etc.).
    - **DELETE /api/habits/:id** — Delete a specific habit.
    - **GET /api/user** — Get user information (name, email, profile).
    - **PUT /api/user** — Update user profile details.
- **Database (MySQL)**:
    - Create a **User Table** (user_id, name, email, password, etc.).
    - Create a **Habits Table** (habit_id, user_id, habit_title, start_date, frequency, status, etc.).
    - Relate users to their habits using **user_id**.

AI Service (Python, Flask, Scikit-learn, Numpy, Pandas)

- Use **Scikit-learn** to build a basic recommendation system.
- Example:
    - If the user's past habits are mostly related to "well-being," recommend habits in that category.
    - Use a simple **rule-based recommendation system** or use basic clustering.
- **Endpoints**:
    - **GET /generate-habit-suggestions** — Return 3 suggested habits.

📋 **Deliverables**

1. **GitHub Repository**:
    - Public/private repo with clean folder structure.
2. **README.md**:
    - Instructions to set up and run the project.
3. **Live Demo (Optional)**:
    - If deployed to **Railway, Render, or Replit**, provide a live link.
4. **API Documentation**:
    - Document API endpoints in README or Postman collection.

🚀 **Submission Instructions**

1. Upload your project to GitHub.
2. Send the **GitHub link** to careers@thrivify.ai.
3. Ensure you include **clear instructions** in the README file.

💡 **Evaluation Criteria**

- **Code Quality**: Clean, modular, and maintainable.
- **Functionality**: Does it work as described?
- **Problem-Solving**: How well you handled the AI recommendation system.
- **Modularity**: Well-structured, reusable components.
- **Deployment (Bonus Points)**: If you deploy the app to a free hosting platform (like **Render, Railway, or Replit**).
- **Consider using Ollama for AI related tasks (You can host it on your machine)**. **Use Llama 2 model.**

🚀 **Pro Tips**

1. **Focus on the Basics**: Get the core functionality right before focusing on styling.
2. **Break Down Tasks**: Work on backend, frontend, and AI separately, then integrate.
3. **Use Free Tools Only**: No paid software to be used.
4. **Add Custom Touches**: Stand out by adding features like habit reminders or streak tracking.

We look forward to seeing your creative solutions! If you have questions, feel free to reach out to careers@thrivify.ai

Good luck and happy coding! 🚀