Full Stack Assessment – Amlgo Labs

Amlgo Labs is hiring a Full Stack Developer, and as part of our selection process, you are required to complete the following technical assessment. The goal of this assessment is to build the project described below, showcasing your fullstack development skills across frontend, backend, databases, and integration.

Project: Personal Finance Tracker+

- ### What You Need to Build:

Create a full-stack web app where users can **track their daily expenses**, **set monthly budgets**, and **view simple reports** to understand how they're spending their money. You'll also create a small **Python service** that gives **smart suggestions** to help users manage their budget better.

- ### What Features Should It Have?

1. User Accounts:

- Allow users to **sign up** and **log in** using email and password.
- After login, each user should only see their own data.
- (Optional): Add an admin user who can see all users' total spending.

2. Add & Manage Expenses:

- Users can **add**, **edit**, and **delete** expenses.
- Each expense should include:
  - **Amount** (e.g., ₹1200)
  - **Category** (e.g., Food, Rent, Shopping, etc.)
  - **Date**
  - **Payment Method** (e.g., UPI, Credit Card, Cash)
  - **Notes** (optional)
- Users should be able to:
  - **Filter** expenses by date, category, or payment method
  - **Search** expenses

3. Budget & Alerts:

- Let users set a **monthly spending limit** for each category.
- Show a **warning or alert** if a category crosses 80% or 100% of its budget.

4. Dashboard (Reports Page):

Create a dashboard that shows:

- Total money spent in the current month
- The category where the user spent the most
- Top 3 payment methods used
- A **pie chart** for category-wise spending
- A **line graph** showing how spending changed over time

You can use any chart library like **Chart.js** or **Recharts**.

5. Smart Suggestions using Python:

- Create a **Python script** or a small **Flask API** that:

  Reads or receives the user's expense data

  - Analyzes spending of the last 30 days
  - Gives smart suggestions like:
    - "You're spending a lot on Food. Try to reduce it by 15%."
    - "Your travel expenses increased a lot this month."
- You can use **Pandas** in Python for this.

(Optional): Return these suggestions in JSON format so they can be shown on the frontend.

6. Monthly Reports using SQL:

- Every month, save a **summary report** in a **SQL database** (like SQLite or PostgreSQL).
- Store things like:
  - User ID, Month, Total Spent, Top Category, Overbudget Categories
- (Optional): Show past 3 months' report using SQL queries

Tools You Can Use:

Part	Tool
Frontend	Next.js (React framework) + TailwindCSS or MUI
Backend	Node.js + Express.js
Main Database	MongoDB (for users, expenses, budgets)
SQL Database	SQLite or PostgreSQL (for reports)
Charts	Chart.js or Recharts
Python	Flask or CLI script + Pandas

- ### Deployment Requirement:

You **must deploy your project** on a hosting platform of your choice:

- Frontend: Use **Vercel**, **Netlify**, or anything similar
- Backend + Python: Use **Render**, **Railway**, or **Cyclic** (or run locally with instructions)

Also share a working live demo link and video along with the GitHub repo when submitting your project.

What You Should Submit:

- A GitHub repository with:
  - Full working code
  - A clear `README.md` that explains:
    - What your app does
    - How to run it (Frontend, Backend, Python)
    - Any test login credentials (e.g., `user@example.com / 123456`)
    - Your live deployed site link
    - Any extra features you added
  - A `.env.example` file that lists all environment variables (do not include secrets)