Full Stack Developer Technical Challenge

Objective:
Build a simple full-stack application that demonstrates your ability to design databases, develop APIs, create a basic frontend, and manage version control.

Part 1: Database Design (ERD & Normalization)

1.
2. Design a simple system for **Expense Tracking** with the following requirements:

3. Each **User** can have multiple **Expenses**.
4. Each **Expense** belongs to a **Category** (e.g., Food, Transport, Bills).
5. Each expense has: amount, description, date, and payment method.

Deliverables:

1. Create an **ERD (Entity Relationship Diagram)** showing all tables and relationships.
2. Normalize the database to **3rd Normal Form (3NF)**.
3. Provide SQL statements to create the tables.
4. Recommended database: **Microsoft SQL Server**.

Part 2: Backend Development (API)

Develop a REST API using **ASP.NET** or **Python (Flask/FastAPI)** that can:

1. Add a new user
2. Add a new expense
3. Retrieve all expenses for a specific user
4. Retrieve total spending per category

Deliverables:

1. API endpoints documented (briefly) in a README file.
2. Include sample requests/responses (can use Postman collection or simple JSON examples).

Part 3: Frontend Development

1. Build a simple **web or mobile UI** using **React, Next.js, or Flutter** that allows:

2. User to view their expenses
3.
4. Add a new expense (form submission to API)

5. Display total spending per category (basic chart or list view)

Deliverables:

1. Functional UI with minimal styling (focus on functionality and logic).

2. Frontend connected to your API.

Part 4: Analytics & Data Insight (Optional Bonus)

Include a simple **summary dashboard** showing:

1. Total spending this month

2. Top 3 expense categories

3. A visual representation (chart or bar graph using any simple library)

Part 5: Version Control & Delivery

1. Initialize a **Git repository** for your project.
2. Use clear **commit messages** for each stage of development.
3. Push your complete project to a **public GitHub repository**.

Include a **README.md** with:

1. Short project description
2. Setup instructions
3. API endpoints
4. Any assumptions or improvements you would add with more time

Evaluation Criteria

Area
Description
Database Design
Proper normalization, relationships, and ERD clarity
Backend
Functionality, clean code, API structure
Frontend
Simplicity, usability, connection to backend
Code Quality
Organization, naming, readability
Version Control
Commit history, structure, and documentation
Optional Analytics
Creativity and use of data insight

Submission

Deadline: **Within 5 days** of receiving this challenge.

Deliverable: **GitHub link** to your repository and, optionally, a short video (max 2 mins) showing your app in action