**📁 Project Overview**

**Name: SmartSpend – Personal Finance Tracker**

**🧱 Key Features:**

* User Registration & Login (JWT-based authentication)
* Add/Update/Delete Expense & Income
* Categorize expenses (Food, Travel, Bills, etc.)
* Monthly summaries & analytics (total spending, highest category, etc.)
* Budget setting & alert if exceeded

**🗃️ Database Design (SQL Server)**

**Tables:**

* Users (UserID, Name, Email, PasswordHash)
* Transactions (TxnID, UserID, Amount, Type[Income/Expense], CategoryID, Date, Note)
* Categories (CategoryID, Name, Type)
* Budgets (BudgetID, UserID, Month, Year, Amount)

**Relationships:**

* One User → Many Transactions
* One User → Many Budgets
* Transactions → Category (many-to-one)

**🔌 Web API Endpoints (ASP.NET Core)**

**Auth APIs:**

* POST /api/register
* POST /api/login

**Transaction APIs:**

* GET /api/transactions
* POST /api/transactions
* PUT /api/transactions/{id}
* DELETE /api/transactions/{id}

**Budget APIs:**

* GET /api/budget/current //representing budget id //
* POST /api/budget

**Analytics APIs:**

* GET /api/summary/monthly
* GET /api/summary/category-wise

**💻 Skills You'll Practice:**

* SQL Server: Joins, Aggregations, Grouping
* ASP.NET Core Web API
* Authentication using JWT
* Clean architecture (Repository Pattern, DTOs, etc.)
* Using Postman or Swagger for testing
* Deploying API on Azure or Render (optional)

**🚀 Resume Boost Points:**

* ✅ Designed and implemented a secure REST API
* ✅ Used SQL Server to design normalized relational schemas
* ✅ Implemented token-based authentication (JWT)
* ✅ Built reporting and analytics endpoints
* ✅ Followed SOLID principles and clean architecture

## ✅ Step 1: Database Schema Design

### 📊 Tables

#### 1. Users

sql

CopyEdit

CREATE TABLE Users (

UserID INT PRIMARY KEY IDENTITY,

Name NVARCHAR(100),

Email NVARCHAR(100) UNIQUE NOT NULL,

PasswordHash NVARCHAR(255) NOT NULL

);

#### 2. Categories

sql

CopyEdit

CREATE TABLE Categories (

CategoryID INT PRIMARY KEY IDENTITY,

Name NVARCHAR(50) NOT NULL,

Type NVARCHAR(10) CHECK (Type IN ('Income', 'Expense')) NOT NULL

);

#### 3. Transactions

sql

CopyEdit

CREATE TABLE Transactions (

TxnID INT PRIMARY KEY IDENTITY,

UserID INT FOREIGN KEY REFERENCES Users(UserID),

CategoryID INT FOREIGN KEY REFERENCES Categories(CategoryID),

Amount DECIMAL(10, 2) NOT NULL,

TxnType NVARCHAR(10) CHECK (TxnType IN ('Income', 'Expense')) NOT NULL,

Date DATETIME NOT NULL,

Note NVARCHAR(255)

);

#### 4. Budgets

sql

CopyEdit

CREATE TABLE Budgets (

BudgetID INT PRIMARY KEY IDENTITY,

UserID INT FOREIGN KEY REFERENCES Users(UserID),

Month INT CHECK (Month BETWEEN 1 AND 12),

Year INT,

Amount DECIMAL(10, 2)

);

## ✅ Step 2: Set Up ASP.NET Core Web API

### 🧰 Tools:

* Visual Studio 2022 or VS Code
* .NET 6 or later
* SQL Server / SSMS
* Entity Framework Core

### 🎯 Project Structure

pgsql

CopyEdit

SmartSpendAPI/

├── Controllers/

├── Models/

├── DTOs/

├── Services/

├── Data/

├── Migrations/

├── Program.cs

├── appsettings.json