Project Title: Personal Finance Tracker

Project Overview:

This project aims to create a web-based Personal Finance Tracker application that allows users to manage their financial activities. Users will be able to input income and expenses, generate reports, and visualize their financial data through a dashboard. The system will provide CRUD functionality for income and expenses and manage user authentication and secure access.

Project Plan Outline

1. Project Goals and Objectives:

- **Main Goal:** Develop a Personal Finance Tracker that allows users to track income, expenses, savings, and generate reports with visualizations.
- **Target Users:** Individuals looking to manage personal finances more effectively.

Key Features:

- Authentication for secure user access.
- CRUD operations for income and expenses.
- Dashboard for an overview of finances.
- Reports and visualizations (pie charts, bar graphs, etc.).
- Secure database to store and retrieve financial data.

---

2. Technologies and Tools:

- **Front-End Technologies:**
    - HTML/CSS for structure and styling.
    - JavaScript/TypeScript for form validation and interactions.
    - React (or Angular/Vue.js) for building interactive components.
    - Bootstrap/Tailwind CSS for responsive design.
- **Back-End Technologies:**
    - C# and ASP.NET Core for server-side logic.
    - Entity Framework Core for database interaction.
    - SQLite or SQL Server for database storage.
- **API Integrations:**
    - Swagger for API documentation.
    - REST API for managing CRUD operations.
- **Development Tools:**

    - Visual Studio for code development.
    - Git for version control and GitHub for project management.
    - Postman for API testing.
    - Swagger for API documentation and testing.

---

3. Major Components Breakdown:

A. Front-End:

- **User Interface (UI):**
    - **Dashboard:** Displays an overview of the user's income, expenses, and savings.
    - **Forms:** To input data for income sources, expenses, and financial activities.
    - **Visualizations:** Using charts to show patterns (pie charts, bar charts, line graphs).
    - **Authentication Pages:** Login, registration, password reset.
- **Frameworks/Tools:**
    - React (or Angular/Vue.js) for building reusable components.
    - Bootstrap/Tailwind CSS for styling.
    - Chart.js or D3.js for data visualizations.

B. Back-End:

- **API and Logic Layer:**
    - Create RESTful APIs for CRUD operations on financial data (income, expenses).
    - Authentication service for user login, registration, password hashing, and validation.
    - Business logic for calculating savings, generating reports, and data filtering.
- **Frameworks/Tools:**
    - ASP.NET Core for building APIs.
    - Identity for user authentication and role management.

C. Database:

- **Schema Design:**
    - **User Table:** To store user credentials and profile information.
    - **Income and Expense Tables:** To store financial data, including date, description, amount, and category.
    - **Reports:** Automatically generated from stored data (optional table).
- **Database Tools:**
    - Entity Framework Core for ORM.
    - SQLite or SQL Server for relational database.

---

4. Development Phases & Milestones:

Phase	Tasks	Duration	Target Completion
Phase 1: Planning & Setup	Define requirements, project setup, design UI mockups, choose tech stack	1 week	MM/DD/YYYY
Phase 2: Front-End Development	Implement UI: Dashboard, forms, authentication pages	2 weeks	MM/DD/YYYY
Phase 3: Back-End Development	Build REST APIs, integrate user authentication, implement business logic	2 weeks	MM/DD/YYYY
Phase 4: Database Design & Integration	Design schema, set up SQLite/SQL Server, integrate with Entity Framework	1 week	MM/DD/YYYY
Phase 5: Testing	Test front-end interactions, back-end API responses, database operations	1 week	MM/DD/YYYY
Phase 6: Finalization & Deployment	Fix bugs, polish UI, deploy app to cloud (optional)	1 week	MM/DD/YYYY

---

5. Features & Implementation Details:

A. Authentication System:

- Users can sign up, log in, and reset passwords securely.
- Passwords stored as hashed values.
- User profile information (First Name, Last Name, etc.).

B. CRUD Operations:

- **Create, Read, Update, Delete (CRUD) for Income & Expense:**
    - Users can add, view, edit, and delete financial records.
    - Each record contains details like amount, category, date, and description.

C. Dashboard & Reports:

- Dashboard displays a snapshot of the user's financial activity (income vs. expenses).
- Reports display visual insights with graphs/charts for spending patterns, income sources, etc.
- **Visualizations:**

    - Pie chart for expense categories.
    - Bar graph for monthly income vs. expenses.

D. API Documentation with Swagger:

- API endpoints documented using Swagger.
- Easily test API functionality through the Swagger UI.

---

6. Risk Management:

- **Technical Risks:** Integration issues between front-end and back-end, delays in third-party API integrations.
- **Mitigation:** Regular testing during development phases, using stubs/mocks for APIs.
- **Timeline Risks:** Unexpected delays or scope creep.
- **Mitigation:** Break down tasks into smaller components, regular progress review with mentors.
- **Security Risks:** Vulnerabilities in user authentication or data storage.
- **Mitigation:** Use hashing (for passwords) and secure APIs (use HTTPS).

---

7. Testing & Review:

- **Unit Testing:** Develop unit tests for business logic.
- **Integration Testing:** Test front-end forms with back-end APIs.
- **User Acceptance Testing (UAT):** Ensure the project meets end-user requirements.
- **Review Process:** Code reviews with mentors, bug fixes, performance optimization.

---

8. Deliverables:

- Fully functioning web application deployed locally or in the cloud.
- GitHub repository with source code and README for setup instructions.
- Final report on the project, including challenges and solutions.
- (Optional) Presentation for demo day.

---

9. Timeline:

- **Start Date:** MM/DD/YYYY
- **End Date:** MM/DD/YYYY
- **Key Milestones:** Phase completion milestones as per the phases outlined above.