Budget

1. Project Overview

- **Goal**: Build a Smart Expense Tracker with C# .NET and Neon Postgres.
- **Features**:
  - User authentication
  - Expense logging
  - Budget management
  - Financial insights
  - Integration with M-Pesa/bank APIs (future)
  - ML for spending trends (future)

2. Tasks

- **Database Setup**:
  - Create tables in Neon.
  - Configure EF Core in .NET.
  - Set up database migrations
  - Implement data seeding
- **API Development**:
  - Create endpoints for users, expenses, and budgets.
  - Implement JWT authentication
  - Add request validation
  - Set up API documentation with Swagger
- **Frontend**:

- Build a simple UI (optional for now).
- Implement responsive design
- Add data visualization components

3. Resources

- **Neon Documentation**: https://neon.tech/docs
- **.NET Documentation**: https://learn.microsoft.com/en-us/dotnet/
- **EF Core Documentation**: https://learn.microsoft.com/en-us/ef/core/

4. Timeline

- **Week 1**: Set up the database and .NET project.
- **Week 2**: Implement user authentication and expense logging.
- **Week 3**: Add budget management and financial insights.

5. Technical Requirements

- **Backend**:
  - .NET 8.0
  - Entity Framework Core
  - PostgreSQL (Neon)
  - JWT Authentication
- **Development Tools**:
  - Visual Studio 2022
  - Git for version control
  - Postman for API testing

6. Security Considerations

- Implement secure password hashing
- Use HTTPS for all API endpoints
- Regular security audits
- Data encryption at rest

7. Testing Strategy

- **Unit Tests**:
  - Service layer testing
  - Controller testing
  - Repository testing
- **Integration Tests**:
  - API endpoint testing
  - Database integration testing

8. Deployment

- Set up CI/CD pipeline
- Configure cloud hosting
- Database backup strategy
- Monitoring and logging setup