Expense Tracker Project Analysis Checklist

---

1. Project Goals and Scope

Details:
The primary goal of the project is to provide a user-friendly app for individuals to track daily expenses, visualize spending patterns, and manage budgets effectively.

Why Important?
A clearly defined scope ensures the app meets user expectations and avoids unnecessary features.

Prompts Used:

1. *"How can I define the core features of an expense tracking app?"*
2. *"What is the ideal scope for an app targeting personal finance management?"*

---

2. User Requirements

Details:
User needs include:

- A simple, intuitive interface for users with varying technical skills.
- Accessibility features like dark mode and scalable fonts.

Why Important?
Understanding user requirements ensures the app delivers real value and enhances the user experience.

Prompts Used:

1. *"What are the primary user needs for a financial tracking app?"*
2. *"How can I design for users with minimal technical knowledge?"*

---

3. Functional Requirements

Details:
Functional features identified:

- Adding and categorizing expenses.
- Setting and updating budgets dynamically.
- Displaying budget overviews and visual charts.

Why Important?
Defining functional requirements ensures the app focuses on delivering core features.

Prompts Used:

1. *"What are must-have features for a budget app?"*
2. *"How do I define functional requirements for a financial application?"*

---

4. Non-Functional Requirements

Details:
Non-functional needs include:

- Quick response times (<1 second).
- Encrypted data for user privacy and GDPR compliance.

Why Important?
Non-functional requirements ensure the app is reliable, secure, and scalable.

Prompts Used:

1. *"What non-functional requirements improve the user experience in financial apps?"*
2. *"How can I optimize app performance for budget tracking?"*

---

5. Technology Stack

Details:
The chosen stack includes:

- **Dart** and **Flutter** for cross-platform development.
- **Firebase** for backend services like authentication and data storage.

Why Important?
The right technology stack ensures scalability, low cost, and maintainability.

Prompts Used:

1. *"What is the best tech stack for building a financial app with Flutter?"*
2. *"How does Firebase simplify backend development for mobile apps?"*

---

6. UI/UX Design

Details:
The design focuses on:

- A clean and easy-to-navigate interface.
- Dark mode and responsive design for accessibility.

Why Important?
Good design makes the app intuitive, attracting and retaining users.

Prompts Used:

1. *"How can I make the UI of my app simple and intuitive?"*
2. *"What are the best UI/UX practices for financial applications?"*

---

7. Data Privacy and Security

Details:

- User data is encrypted with Firebase rules.
- OAuth2 authentication ensures secure login.

Why Important?
Privacy and security build trust, which is essential for financial apps.

Prompts Used:

1. *"How can I ensure secure data handling in a Flutter app?"*
2. *"What are the best practices for implementing authentication in financial apps?"*

---

8. Integration with Third-Party Tools

Details:

- Google Sheets API for exporting budgets.
- Google Analytics for tracking user behavior.

Why Important?
Third-party tools enhance the app without heavy in-house development.

Prompts Used:

1. *"What third-party tools are helpful for managing financial data?"*
2. *"How can I integrate Google Sheets with a Flutter app?"*

---

9. Testing Strategy

Details:

- Unit tests for budget calculations.
- Integration tests for seamless navigation.

Why Important?
Testing ensures the app is stable and reliable under various conditions.

Prompts Used:

1. *"What testing strategies are most effective for a Flutter app?"*
2. *"How do I write effective integration tests for financial apps?"*

---

10. Maintainability and Scalability

Details:

- Modular code structure with reusable components.
- Clean coding practices (DRY, KISS).

Why Important?
Maintaining a clean codebase reduces technical debt and supports scalability.

Prompts Used:

1. *"How can I structure my code for long-term maintainability?"*
2. *"What clean code principles are essential for scalable Flutter apps?"*

---

11. Accessibility

Details:

- Scalable fonts and proper semantic labeling.
- Dark mode toggle for better usability in low-light environments.

Why Important?
Accessibility ensures inclusivity, making the app usable for people with disabilities.

Prompts Used:

1. *"What accessibility features can enhance a Flutter app?"*
2. *"How can I make my app more user-friendly for visually impaired users?"*

---

12. Performance Optimization

Details:

- Caching for frequently accessed data.
- Optimizing widget rebuilds using Provider.

Why Important?
Optimized performance prevents app crashes and ensures a smooth experience.

Prompts Used:

1. *"How can I reduce widget rebuilds in Flutter apps?"*
2. *"What techniques improve performance for real-time financial apps?"*

---

13. Deployment and Distribution

Details:

- A CI/CD pipeline for automated testing and builds.
- App Store and Play Store optimization for higher discoverability.

Why Important?
Streamlined deployment accelerates time-to-market.

Prompts Used:

1. *"What is the best way to set up a CI/CD pipeline for Flutter apps?"*
2. *"How can I optimize my app for both Play Store and App Store?"*

---

14. User Feedback Integration

Details:

- In-app feedback forms.
- Analytics to track feature usage and pain points.

Why Important?
Feedback helps prioritize updates and improve user satisfaction.

Prompts Used:

1. *"What's the best way to gather in-app user feedback?"*
2. *"How do I use analytics to improve app usability?"*

---

15. Metrics for Success

Details:
Metrics tracked include:

- Daily active users (DAU).
- Retention rates after 30 days.
- Session duration for engagement tracking.

Why Important?
KPIs measure the app’s success and inform future improvements.

Prompts Used:

1. *"What KPIs should I use to evaluate the success of a financial app?"*
2. *"How can I monitor user engagement effectively in Flutter apps?"*