PiggyNest Project To-Do List

# 1. Initial Setup

- Set up project directory and create a version control repository (e.g., Git).

- Choose technology stack (Frontend: React.js, Backend: Node.js, Database: MongoDB or PostgreSQL).

- Install necessary libraries and dependencies.

# 2. Homepage

- Design and create the homepage layout.

- Create 5-6 sections with clear content (e.g., About Us, Features, Contact, etc.).

- Implement up to 4 navigation links (e.g., Home, Dashboard, About, Contact).

- Ensure the homepage is responsive for desktop and mobile.

# 3. User Authentication and Validation

- Design and create login page with email and password input fields.

- Implement backend functionality for user authentication (using JWT or sessions).

- Add user validation (e.g., password strength, unique email).

- Handle errors (invalid login, empty fields).

- Add “forgot password” functionality.

# 4. Dashboard

- Create the main dashboard layout.

- Add sidebar with 5 functional links (e.g., Overview, Transactions, Settings, Profile, Logout).

- Ensure responsive design for the dashboard.

- Design and create a user-friendly data display (e.g., transaction history, balance overview).

# 5. Transaction Management

- Design and implement transaction management page (Create, View, Edit Transactions).

- Implement backend functionality to handle transactions (e.g., transfer, deposit, withdrawal).

- Create transaction history display (with sorting and filtering options).

- Add form validation for transaction creation.

- Ensure security measures for transaction processing (e.g., two-factor authentication, encryption).

# 6. Testing

- Test the homepage for layout and functionality.

- Test login authentication (valid and invalid login attempts).

- Test dashboard navigation and responsiveness.

- Test transaction management for accuracy and error handling.

- Perform integration testing (Frontend and Backend).

# 7. Deployment

- Set up the web server (e.g., deploy on Heroku, AWS, or DigitalOcean).

- Deploy the frontend (React app) and backend (Node.js API).

- Configure the domain and ensure HTTPS for secure communication.

# 8. Future Enhancements

- Plan for adding more advanced features (e.g., loan management, investment options).

- Prepare for future conversion to desktop app using Electron.js.