Sure, here are some features you might consider for a personal finance tracker:

1. **Account Management**: Allow the user to create and manage multiple accounts (like checking, savings, credit card, etc.).
2. **Transaction Logging**: Provide a way to log income and expenses. Each transaction should include the date, amount, category (like groceries, utilities, entertainment, etc.), and a note field for additional details.
3. **Budgeting**: Allow the user to set a budget for each category and track spending against the budget.
4. **Reports and Analysis**: Provide reports that show spending by category and over time. This could include pie charts, line graphs, etc.
5. **Recurring Transactions**: Allow the user to set up recurring transactions for bills that are the same amount and due on the same day every month.
6. **Alerts and Reminders**: Send alerts when the user is close to exceeding their budget, and reminders for upcoming bills.
7. **Data Import/Export**: Provide a way to import transactions from bank statements (like CSV files) and export data for backup or use in other applications.
8. **Security**: Since financial data is sensitive, make sure to implement strong security measures. This could include encryption for data storage and transmission, a strong password policy, and possibly two-factor authentication.

Here's a rough plan for how you might implement this:

1. Define your database schema. You'll need tables for users, accounts, transactions, categories, and possibly others depending on your features.
2. Create your Laravel models, migrations, and controllers for each of these tables.
3. Implement the account management features. This includes creating, viewing, updating, and deleting accounts.
4. Implement the transaction logging features. This includes creating, viewing, updating, and deleting transactions.
5. Implement the budgeting features. This includes setting a budget for each category and tracking spending against it.
6. Implement the reports and analysis features. This could involve creating a separate reports controller, or it could be part of the transactions or categories controller.
7. Implement the recurring transactions, alerts and reminders, and data import/export features.
8. Implement the security features. This includes encrypting sensitive data, enforcing a strong password policy, and possibly implementing two-factor authentication.
9. Test your application thoroughly to make sure everything works as expected.
10. Deploy your application to a web server.

Remember to start small and gradually add more features. It's better to have a simple app that works well than a complex app that's full of bugs. Good luck with your project!