Updated Features

1. **User Authentication**
    - Basic signup/login.
2. **Transaction Management**
    - Users can log income and expenses manually.
    - Add receipts by uploading images (integrate a free OCR API like OCR.Space or Mindee).
    - Ex: Someone puts in the income and all the expenses in the budget and then its suggests a weekly budget. Throughout the week the person can upload receipts and manual entries. Each transaction will subtract from the budget displayed on the dashboard, which will have a visual representation of the highlights of your finances when you first open the app.
3. **Budgeting**
    - Users can set monthly budgets by category.
        i. Income
        ii. Rent
        iii. Car insurance
        iv. Phone bill
        v. ect
4. **Currency Conversion**
    - Convert transactions to different currencies. (Chat gbt rec)
5. **Dashboard Visualization**
    - Show spending trends, category breakdowns, and budget tracking via interactive charts.

Receipt Scanner Integration

- **API**: Use OCR.Space (free tier) for extracting text from receipt images.
- **Workflow**:
    1. User uploads a receipt image.
    2. The image is sent to the OCR API.
    3. Extracted text is parsed to identify transaction details (e.g., amount, category, date).
    4. Transaction data is stored in the database.

Database Schema

1. **Users**:
    - user_id (Primary Key)
    - email
    - password_hash

2. **Transactions**:
    - transaction_id (Primary Key)
    - user_id (Foreign Key)
    - amount
    - category (e.g., food, rent)
    - currency
    - date
3. **Budgets**:
    - budget_id (Primary Key)
    - user_id (Foreign Key)
    - category
    - limit
    - period (e.g., monthly, yearly)

Dashboard Visualization

Use **Chart.js** to create interactive spending dashboards.

Example Dashboard Features:

1. **Spending by Category**:
    - Pie chart to display spending distribution across categories.
2. **Spending Trends**:
    - Line chart showing spending trends over time.
3. **Budget Progress**:
    - Bar chart to show how close users are to exceeding their budget.

Updated 3-Week Timeline

Week 1: Planning & Setup

- Define epics and stories in Trello.
- Set up GitHub repository.
- Create database schema.
- Build static HTML/CSS pages for login, transaction form, and dashboard.

Week 2: Core Development

- Backend:
    - Create APIs for user authentication, transactions, and receipt uploads.
    - Integrate the OCR API for receipt scanning.
- Frontend:
    - Add interactivity to forms with JavaScript.
    - Set up Chart.js for dashboards.
- Database:
    - Populate test data and integrate with the backend.

Week 3: Integration & Testing

- Integrate frontend and backend.
- Test receipt uploads and transaction logging.
- Fine-tune the dashboard visualizations.
- Prepare for the demo and finalize documentation.

Deliverables

1. **Functional App**:
    - Login/signup.
    - Add transactions manually or via receipt upload.
    - Dashboard with interactive charts.
2. **Database**:
    - MySQL database populated with sample data.
3. **Documentation**:
    - README with setup instructions.
    - Demo script for presentation.