

Final Project Planning

**** Stretch goals are marked with a ****

Tech Stack:

- React (options: Reach UI to create modal tabs)
- Express, Node.js
- PostgreSQL
- Graphing libraries (react-charts, charts.js?)

User Stories:

1. Anyone can view the **home page** to see the features the app offers - summary of the app
2. Anyone can **register**
3. Registered users can **log in through the nav bar on the home page**
4. Logged-in users can view the **dashboard** (summary of total balance, income and expense summary chart, monthly balance chart, and summary of transactions (5-10), ****budgets**** and ****recurring transactions****)
5. Logged-in users can navigate to various pages using the left-side menu
6. Logged-in users can see all **transactions** in reverse chronological order on the transactions page, and can filter transactions by month
7. Logged-in users can add/edit transactions or add/edit transfers and delete transactions through the transactions page
8. Logged-in users can see the **accounts** page
9. Logged-in users can add new accounts (modal) and delete current accounts through the accounts page
10. Logged-in users can see the **reports** page (with charts for total expenses and income by category)
11. Logged-in users can see the **categories management** page
12. Logged-in users can add/edit/delete categories if they the preset categories are not applicable
13. ******Logged-in users can see the **budgets** page, and a progress bar for the percentage of the budget they have already used up******
14. ******Logged-in users can add new/edit budgets based on categories******
15. ******Logged-in users can see new **recurring transactions** page******

16. ******Logged-in users can add new recurring transactions on the recurring transactions page******
17. Logged-in users can log out and return to home page
18. ******Allow users to toggle between light and dark modes******
19. ******Make the application responsive (desktop, and mobile compatible)******

Wireframe:

Database Tables:

users		DB values
PK	ID	SERIAL PRIMARY KEY not null
	name	VARCHAR(255) not null
	password (hashed/salted)	VARCHAR(255) not null
	e-mail	VARCHAR(255) not null

transactions		DB values
PK	ID	SERIAL PRIMARY KEY not null
FK	user_id	INTEGER REFERENCES users(id) ON DELETE CASCADE
FK	category_id	INTEGER REFERENCES categories(id) ON DELETE CASCADE
FK	account_id	INTEGER REFERENCES accounts(id) ON DELETE CASCADE
	amount	INTEGER not null DEFAULT 0
	date	TIMESTAMP not null DEFAULT NOW() □ suggestions on how to do this?
	recurring?	BOOLEAN not null DEFAULT FALSE
	notes	TEXT

accounts		DB Values
PK	ID	SERIAL PRIMARY KEY not null
FK	user_id	INTEGER REFERENCES users(id) ON DELETE CASCADE

	account_name	VARCHAR(255) not null
	balance	INTEGER not null DEFAULT 0

categories		DB Values
PK	ID	SERIAL PRIMARY KEY not null
	category_name	VARCHAR(255) not null
	type (income or expense)	VARCHAR(255) not null
FK	user_id	INTEGER REFERENCES users(id) ON DELETE CASCADE
	logo_url	VARCHAR(255)

budgets		DB Values
PK	ID	SERIAL PRIMARY KEY not null
FK	user_id	INTEGER REFERENCES users(id) ON DELETE CASCADE
FK	category_id	INTEGER REFERENCES categories(id) ON DELETE CASCADE
	budget_amount	INTEGER not null DEFAULT 0

recurring_transactions		DB Values
PK	ID	SERIAL PRIMARY KEY not null
FK	user_id	INTEGER REFERENCES users(id) ON DELETE CASCADE
FK	category_id	INTEGER REFERENCES categories(id) ON DELETE CASCADE
FK	account_id	INTEGER REFERENCES accounts(id) ON DELETE CASCADE
	amount	INTEGER not null DEFAULT 0

	date	TIMESTAMP not null DEFAULT NOW()
	repeats_every	VARCHAR(255) not null
	number_of_times	INTEGER not null DEFAULT 0
	notes	TEXT

React components: (SPA recommended)

- App.jsx (default with create-react-app)
- HomePage.jsx
 - Login modal
- Dashboard.jsx
- Transactions.jsx
 - Add new transactions modal (by clicking + float button → 'Transactions' tab)
 - Edit transactions modal (via 3 dots on particular transactions card)
 - Add new transfers modal (by clicking + float button → 'Transfers' tab)
 - Edit transfers modal (via 3 dots on particular transactions card)
- Accounts.jsx
 - Add new account modal (by clicking + float button)
- Reports.jsx
- Categories.jsx
 - Add new categories modal (by clicking + float button)

Routes:

- / → home page/main page/landing page
 - GET
- /login → login page
- /register → register page (can add a register link in the login modal to avoid this route)*
- /:user/overview → dashboard page
 - GET
- /:user/transactions → transactions page
 - GET
- /:user/transactions → add new transaction (through modal)

- POST
- `/:user/transactions/:id/delete` → delete a transaction
 - POST
- `/:user/transactions/:id/edit` → edit an existing transaction (through modal)
 - POST
- `/:user/accounts` → accounts page
 - GET
- `/:user/accounts` → add new account (through modal)
 - POST
- `/:user/accounts/:id/edit` → edit an existing account (through modal)
 - POST
- `/:user/accounts/:id/delete` → delete an existing account
 - POST
- `/:user/report` → report page
 - GET
- `/:user/category` → category management page
 - GET
 - POST
- ****** `/:user/budgets` → budgets page ******
 - GET
 - POST
- ****** `/:user/recurring` → scheduled/recurring page ******
 - GET
 - POST
- `/logout` → return to home page *****
 - GET