Developer Requirement Analysis

Components Required

Database: mongoDB, MongoDB Atlas,

API: express.js, Mongoose

Central Data Store: Redux,

Frontend: react.js, chart.js

Details

1. Mongodb will be used as base database software. Mongodb Atlas will be used to have an online database server by having a reserved cloud cluster. Mongoose will be used to connect the Atlas database service to the node project. Mocha will be used to test the database with APIs.
2. Express.js will be used to create APIs to perform CRUD operations on the database.
3. Redux is used to create a central data store so that the components: calculator and grapher could CRUD the data. As syntax it would have its own dispatchable actions and a reducer to execute those actions. Dispatched actions would be executed by Redux using the APIs created in the second step.

TBD: Explore Firebase.

1. Uses React. Frontend mainly consists of Home, Dashboard and Log/Sign In. Dashboard consists of Calculator and Grapher. Calculator would read data, edit rows, delete rows and create rows. Grapher could only read data.

TBD: Definitely use one of the following: Fusion charts, chart.js for Grapher.

Rough Model

Frontend Isolatable Content + Frontend Buttons and Non-Isolatable Content (Communicate with Reducer by dispatching actions)

{ Use redux central store to react to dispatched actions (communicate with APIs)

{ Create and use express APIs to apply CRUD operations on database

{database}

}

}

Database Schema

Name of Database: budgeter

Types of Collections(Tables): users, user\_details

users schema:

{

\_id: ObjectID,

username: String,

email: String,

password: Encrypted\_String,

first\_name: String,

last\_name: String,

register\_date: Date()

}

user\_transactions schema:

{

\_id: ObjectID,

user\_id: ObjectID,

user\_economy: Array()

{

\_id: ObjectID,

group: String,

title: String,

description: String,

amount: Number,

createdAt: Date()

updatedAt: Date()

}

}

Webpages

Navbar

Title on left, Dashboard and Login on right.

Home

Big Title with Big Background Image, Detailed Description, Sign In/ Log In, Manual or How to use.

Sign In

Email and password options, sign in button

Sign Up

First name, last name, username, email, password, create button

DashBoard

Having the option to switch between “Budgeter” and “Analytics”.

Budgeter

Title at the top.

Having a timeline on the left side to switch between months to edit tables.

Table consists of a header at top, rows of data in middle and “add new” at the bottom.

Rows consist of collapsible horizontal bars showing the group name and total income/expenditure. Clicking on the bar can collapse and uncollapse it. Uncollapsed version shows all the added rows with date, title and amount.

“Add new” button opens a note that asks the user to fill in the necessary fields: title, description, amount and group.

Total Income, total expenditure and savings of the month is shown at the bottom.

Analytics

Title at the top.

Graph in the middle. Graph shows the area chart (review) of all the groups.

Timeline at the bottom. Could be used to view either: individual months, years or overall.

Routes

Client Accessible:

‘/’ : Home Page

‘/auth’ : Sign Up : REGISTER

‘/auth’ : Sign In : SIGN IN

‘/dashboard/budgeter’ : Budgeter : CRUD

‘/dashboard/analytics’ : Analytics : GET

Server Routes:

‘/user/admin’ : GET, POST

‘/user/admin/:id’ : DELETE

‘/user/signin’ : AUTH

‘/user/signup’ : POST

‘/dashboard/admin’ :GET ALL

‘/dashboard/admin/:userId’ :GET ONE

‘/dashboard/:userId/analytics’ :GET

‘/dashboard/:userId/budgeter’ :GET, PATCH(ADD)

‘/dashboard/:userId/budgeter/:id’ :PATCH(UPDATE), PATCH(DELETE)