

EZ Budget

Contributors:

Diti Gupta, Logan Brown, Jillian Bartz, Rishi Hancock, Adam Fox, Will Davidson

Description:

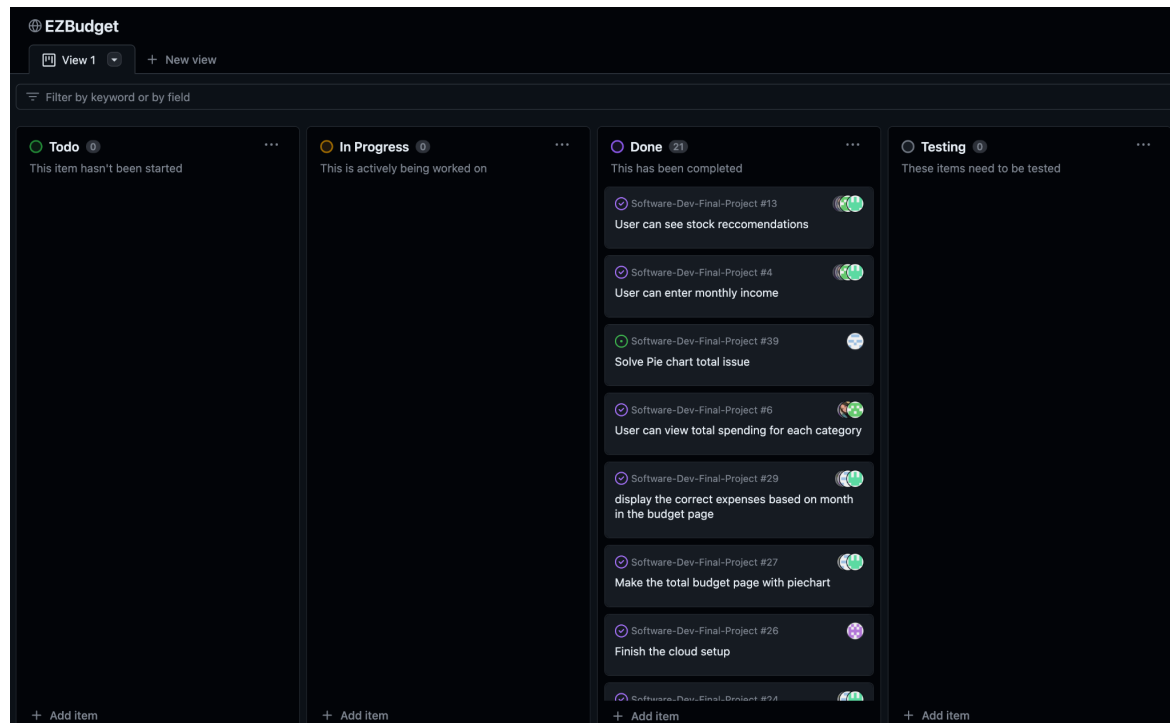
EZBudget is an application that allows users to track their monthly income and allocate spending to assist users in managing their monthly and yearly budgets. The application will allow users to create an account that will store their personal budget information which can be accessed through the login page.

EZBudget will ask users for monthly incomes, as well as any spending that the user wants to include. There will also be individual categories for spending, including transportation, investments, transportation, and other monthly expenses. The application will provide the total sums of their personal spending for each category on their budget tab, as well as their net profit for each month and year to date.

The EZBudget application is extremely useful for any individuals looking to become more aware of their spending habits. This application can be a quick resource for individuals to see their budget without having to independently compute their expenses or incomes. EZBudget will provide simple information regarding the user's categorical spending habits, allowing for more opportunities to successfully manage their finances in the future months. The application will include monthly tabs that can show the user's overall progress as well as a yearly tab that can show the user their total expenses and income.

Project Tracker:

<https://github.com/users/willdavidson05/projects/1/views/1>



Video:

[EZBudget Demo Video](#)

VCS:

<https://github.com/willdavidson05/Software-Dev-Final-Project>

Contributions:

Logan - Worked with team members to design and deploy the database used in the backend. Also helped with functions such as add /addexpense and others to pass data from the front end to the database.

Diti - Worked on backend work with addExpense/deleteExpense functions as well as updating each month page. Additionally, worked on the pie charts for each month page. Helped with deployment on Azure.

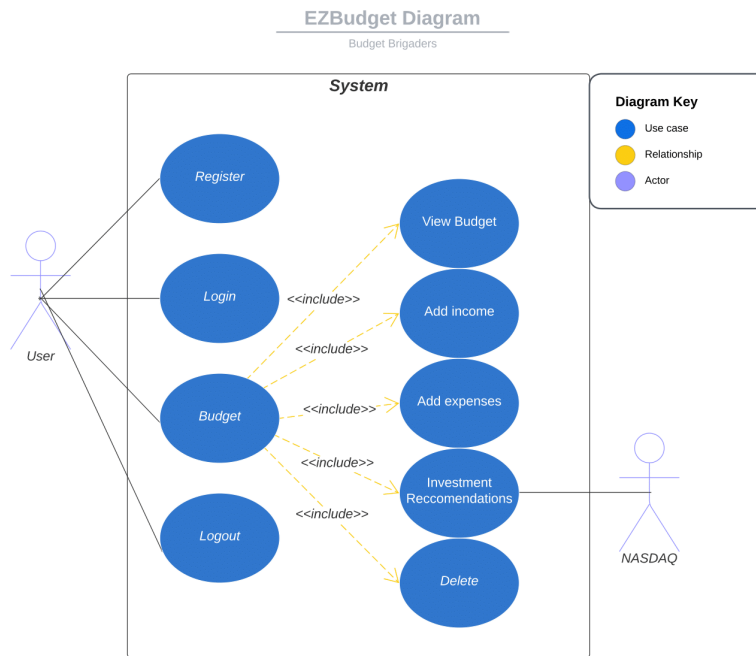
Adam - Helped alongside team members to design and deploy the database, assisted in creating ejs files for each month and additionally added an interactive breakdown to display the monthly breakdown of expenses.

Rishi - Helped with test cases and deployment on Azure.

Will - Worked on the design and implementation of the database used for the backend. Also worked on the addExpense/deleteExpense to correctly update monthly data.

Jillian - Developed and designed the front end of the website, including the CSS styling sheet for the home, login, register, budget, and logout pages. Assisted in creating ejs tabs and structure of website wireframes.

Use Case Diagram:



Test Results:

Test Case #1:

Register Test case:

- Ensure the user can register a new account using a unique string username and unique email.

Observations:

- The user inputs a username and password into the input sections of the register page. If the username exists, a message pops up: "User already exists", and the user is then able to navigate to the login page instead where they will be able to log into their already existing account. If the user is not registered, after inputting the username, and password, the user is redirected to the login page where they can now login into their new account.
- In general, the behavior is consistent with the use case, and there is no deviation from the expected actions.

Test Case #2:

Login Test case:

- Ensure that the user can login successfully after creating a new account or login into a previous account.

- Positive test case: Check that the user inputs for username/password match with data from the users table from the database. If the inputs for username/password match with the database username/password, redirect the user to the user's budget page, and return the 200 status.

Observations:

- The user inputs a username and password into the input sections of the login page. If the username exists in the database, check if the hashed password entered by the user matches the hashed password stored in the database. If the username and password match, log the user in and store their session. If the username entered does not exist in the database, redirect the user to the register page instead. If the user input for username matches but the hashed password does not match the stored password from the database give an error message: "Incorrect password"
- In general, the behavior is consistent with the use case, and there is no deviation from the expected actions.
- In general, the behavior was consistent with the use case, and there was no deviation.

Deployment:

<http://recitation-12-team-04.eastus.cloudapp.azure.com:3000/>