

BudgetBeam

Group Members: Patrick Hunt, Michael Thomas, Andrew Yang, Julie Huang

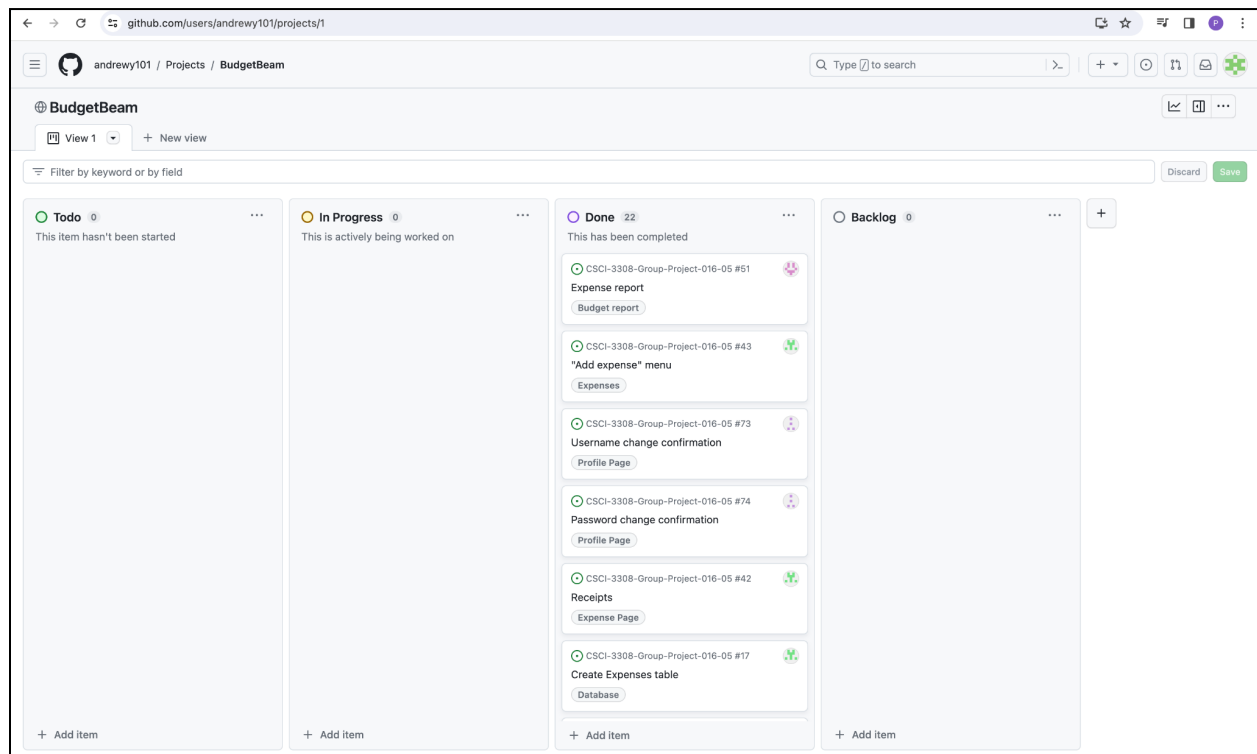
I. Project Description

BudgetBeam is an expense tracking application which utilizes PostgreSQL, Docker, NodeJS, and ExpressJS. The team focused on creating an application that is as easy to use as possible so that users interested in entry-level financial management do not feel intimidated and/or overwhelmed by trying to manage their finances, as many expense tracking applications present too much information to the user, causing unnecessary confusion. BudgetBeam's functionality includes allowing users to add receipts for different expenses/incomes and aggregating all of this data into a presentable, concise dashboard for their account and breaking down expenses by month and category, allowing users to assign budgets to specific categories for any given month. Users can also generate expense reports for the month of their choosing which can then be converted into a downloadable PDF if the user chooses to do so, which can then be kept as a financial record for the user. BudgetBeam also allows users the flexibility to customize their account by changing their username and/or password, and allowing for account deletion if desired. The combination of these functionalities makes BudgetBeam a useful tool for tracking expenses while still upholding the user-friendly aspect of the application that the team had initially strived for.

II. Project Tracker (GitHub Project Board)

- Link to Project Tracker: <https://github.com/users/andrewy101/projects/1>

Screenshot of Project Tracker



III. Video Demonstration

- Link to shared google drive with video demo:

https://drive.google.com/drive/folders/18JVdr4knnc4XokAd8LrEuIU_e40E9ZdK?usp=drive_link

IV. VCS

- Link to GitHub Repository:

<https://github.com/andrewy101/CSCI-3308-Group-Project-016-05>

V. Contributions

Patrick

- For this project my primary role was to work on the homepage feature which displays all of the user's expense/income information broken down by category and month. This included working on all of the endpoints/backend components for the homepage functionality using NodeJS and PostgreSQL, and EJS. I also worked on the UI for the homepage, utilizing technologies such as HTML, Bootstrap, and CSS, and I also utilized CanvasJS to create a dynamic pie chart. I was responsible for writing the test cases for the project using Mocha and Chai frameworks to ensure that our code was always functioning as it should be.

Andrew

- Implemented the frontend and backend for the receipts page. Implemented user session authentication. Finalized layout for the navigation bar and login/registration pages.

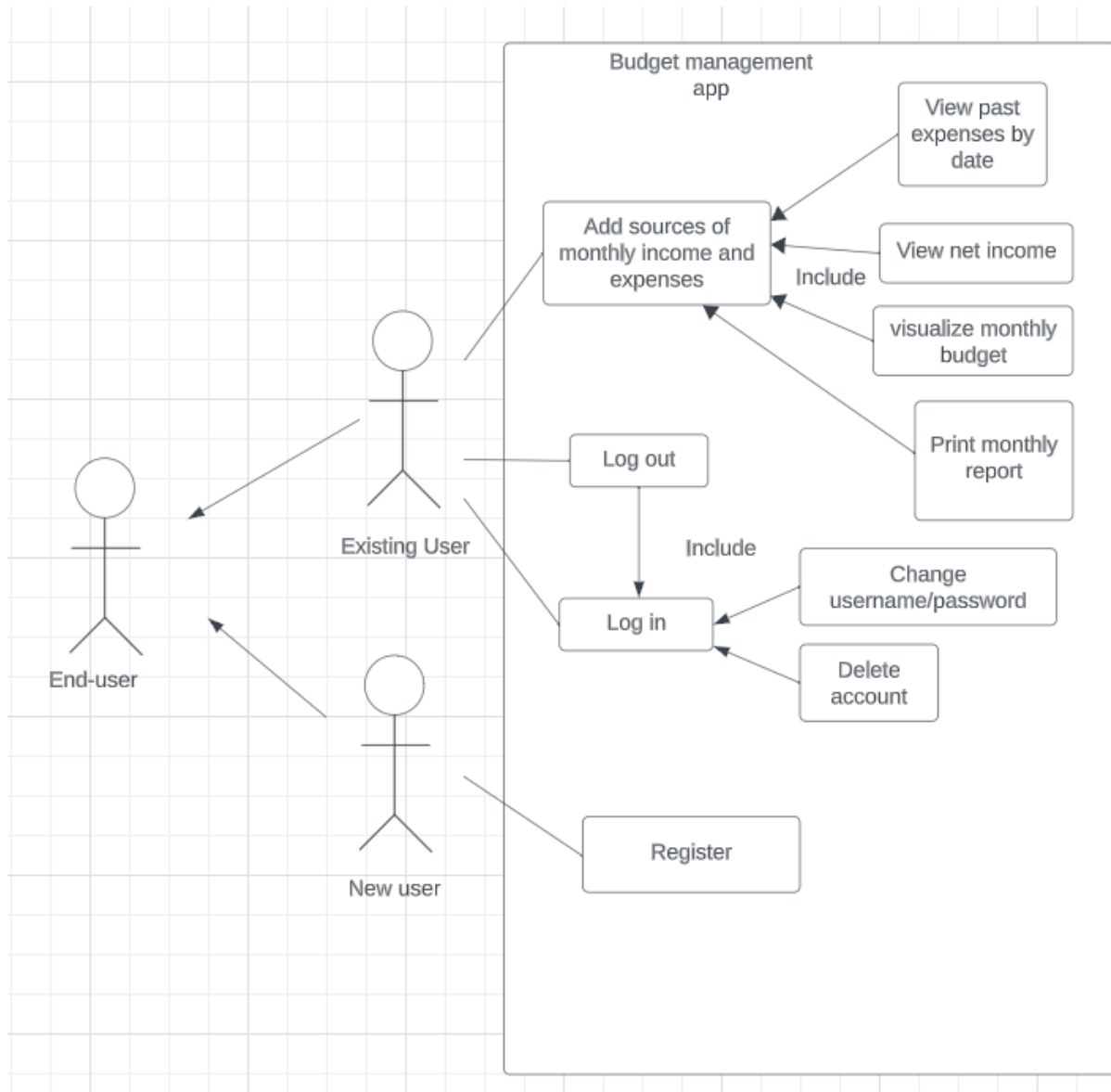
Michael

- Created the report page and added pdf download functionality. Reviewed code. Worked on various project documents and diagrams.

Julie

- Developed the profile settings page with user ability to change username, change password and delete their account as desired. Finalized presentation slides.

VI. Use Case Diagram



VII. Test Results

- Test 1 - Login page
 - Description
 - In this test case, the user will be instructed to log into our web application using credentials for an existing user. This entails navigating to the login page of the website, entering the appropriate data in the appropriate fields in the login form, and submitting the form by pressing the 'Login' button. Upon success, the user will be redirected to the homepage.
 - Observations
 - The user started by visiting the appropriate URL in the browser, in this case 'http://localhost:3000/login'.
 - The user then immediately navigated to the individual username and password fields without hesitation.
 - The user filled out the login form correctly and knew to press the 'Login' button when they were done, because the button is clearly labeled.
 - The users actions were consistent with this use case and there were no deviations from the expected actions.
 - Because of the success of this use case, we did not modify the login functionality for our application.
 - Results:
 - The test for this use case was successful as the user was able to properly log into our application with no issues and were successfully redirected to the homepage, and they followed the expected steps to be taken in order to login.
- Test 2 - Creation of new expense
 - Description
 - In this use case, the user needs to add a new receipt/expense to their account after logging into the application. This includes navigating to the 'Expenses' page from the menu, clicking on the 'Add Receipt' button, and then filling out the appropriate fields within the form and submitting the form by clicking on the 'Create' button. Upon success, the user will be able to see their new expense.
 - Observations
 - After logging in, the user was redirected to the homepage as expected, but instead of navigating to the 'Expenses' page on the menu, they first clicked on the 'Expense Report' page. Their reasoning for this was that they simply saw the word 'Expense' before reading 'Report' so they just assumed that it was the correct page. This seemed like an isolated incident, and it did not hinder the overall process of creating a new expense enough to justify renaming this menu item, so no changes were made.
 - After this, the user completed the task as expected. They corrected their mistake with ease and navigated to the correct 'Expenses' page. They then

clicked on 'Add Receipt', filled out the form correctly, and submitted the form.

- There were no serious deviations from the expected actions for this use case aside from first navigating to the incorrect tab.
- Results
 - The test for this use case was successful as the user was properly able to add a new expense and the user was given confirmation that this operation was successful as they were able to see their new expense appear on the expense page.
- Test 3 - Generation of expense report
 - Description
 - In this use case, the user needs to generate an expense report for a month of their choosing. This includes navigating to the 'Expense Report' page after logging in and selecting the appropriate month from the 'Month' dropdown menu. Upon success, the user will be shown an expense report for the selected month in table format, and will be given the option to generate a PDF of that report.
 - Observations
 - After logging in, the user first navigated to the correct tab on the menu ('Expense Report'). They first attempted to generate an expense report for months that they did not have expenses for yet, their reasoning being that they didn't immediately remember which month they had expenses for, but we displayed a message on this webpage to accommodate for this issue that clearly indicates that there is no data for the selected month, so the user quickly corrected this issue and selected the appropriate month from the dropdown.
 - There were really no deviations from the expected actions for this use case, we did not consider selecting a month with no expenses yet a 'deviation', because we anticipated this issue and handled it appropriately by displaying a message, which proved to be effective during this test because the user was able to correct their mistake because of it.
 - The user decided to generate a PDF for their expense report after selecting the appropriate month by clicking on the 'Download PDF' button, and was successfully able to view the PDF.
 - We considered the user's actions during this test to be consistent with the use case.
 - Results
 - The test for this use case was successful, as the user was able to successfully view an expense report with the correct data corresponding to the month that they selected. They were also given the option to download a PDF for the report as expected, which they then did. Our warning message to handle the case when there are no expenses for a selected month proved to work as we expected it to and allowed the user to correct their choice of month.

VIII. Deployment

- Link to Deployed Application:

<http://recitation-16-team-05.eastus.cloudapp.azure.com:3000/>