‘BUDGET BUDDY’

Project Documentation

Group 2: Jessie, Chahat and Clarisse.

List of Contents

Introduction

Aims and Objectives:

The original aim, as laid out in the first assignment handout, was to create an innovative travel and budgeting website to help travellers plan their trips more efficiently and economically, including a feature that automatically calculates and distributes shared expenses among travel group members. This seeks to resolve some of the worries of keeping on top of all the group travel costs.

Jessie put forward the budgeting app idea at the start of our planning. We were all happy to go ahead with building this model and spent subsequent meetings designing the code layout together. We organised ourselves with weekly meetings, and regular contact across a shared GitHub repository, the slack channel, a shared notes board, Trello, a log of meeting minutes and objectives, and an individual activity log*.

We each completed SWOT analyses which helped us delegate ownership over distinct aspects of the project. Jessie primarily focussed on designing the HTML templates and CSS files, while Clarisse and Chahat wrote the python code and SQL script. We also approached the work with an agile methodology, working together flexibly and collaboratively. Importantly, this experience pushed us to welcome changing requirements late in development, as time constraints and a smaller group size resulted in us pivoting our application to the python console, rather than webapp. This will be discussed further in the specifications and designs portion of this report.

**See appendix 1 for a copy of the Activity Log with a breakdown of our individual contributions.*

Roadmap of the Report:

This report will begin by briefly providing general background to the details of the chosen topic, and a simple summary of the intended flow of the code. This will lead into a more in-depth breakdown of the specifications and design, which will account for the decision to push the HTML into a development branch and focus on the console application.

We will then expand on how we approached this project as a team in the Implementations and Execution section of this report. Finally, we will explain how we tested some of the functionality of our webapp, and any system limitations we have reflected on during this project.

Background

Our project is intended to be used as a webapp.

Originally, we were working towards completing a functional webapp through flask in python which would render HTML templates and pass variables between the webapp and the database.

We intended to allow users to:

- Login/ Create Account
- View Trips/ Add Trip
- View Costs/ Add Contributions
- Divide Costs and Contributions
- Acknowledge Payment Milestones

As a group, we have adapted to time constraints, group illness, and missing a group member to ensure we had a working application to meet our deadline. Therefore, we have chosen to present a python console simulation of the webapp, along with a mock data demonstration of the HTML templates showcasing the working API connection. In a real-life scenario, this would enable us to present a working product, showcase the potential of the product, and continue to work to upgrade the project into a webapp. Similarly, we streamlined the functionality of the console to demonstrate proof of concept in time for the deadline.

The python console allows users to:

- Login/Create Account
- View Trips/ Add Trips
- Add Contributions

With this in mind, the python console is designed to mimic the functionality of the HTML templates. This will be elaborated on in the following Specifications and Designs portions of the group report.

Specifications and Designs

Functional vs Non-functional requirements:

Functional Requirements:

- User Authentication.
- Database Interaction.
- Frontend Integration.
- Testing.
- Project Management.

Non-functional Requirements:

- Security: hashing password, storing 'session' ids.
- Usability: make webpages and console design easy to follow.
- Scalability: develop webapp, scale to multiple users in same trips.
- Maintainability: Well annotated code, clear and regular git commit messages.

Design and architecture:

The design and architecture of this group project is easier to understand through flow charts.

1) We began by designing the SQL table.
2) We developed our Webapp layout.
3) We translated the Webapp to a python console.

**See appendix 2 for a copy SQL design.*

**See appendix 3 for a copy of the webapp design.*

**See appendix 4 for a copy of the python logic.*

The important thing to take note of is that the console app has been designed as a placeholder for the webapp. This is reflected in the way in which the files are organised.

- 'main.py' acts as the launching file, as 'app.py' did in the webapp.
- All the templates folder HTML files have been given equivalent python files to simulate the user experience and python-SQL functions and logic.
- Rather than user hyperlinks, or url_redirect in flask, the console app utilises recursive functions to adapt for the procedural nature of python console scripts.

Implementations and Execution

Development Approach and Team Member Roles:
We used a very collaborative approach to development, making use of each team member's unique abilities to guarantee a productive workflow. This has been discussed in the introduction.

Tools and Libraries:
To speed up development, we used a number of tools and libraries, including:

- Flask: For building the user-related services and API endpoints for authentication on the web framework.
- User credentials and authentication information are stored in MySQL, a relational database.
- unittest & mock: This refers to mimicking database calls (query_db and insert_db) and testing authentication routines in unit testing.
- Hashlib: This tool ensures that private data is safely stored by hashing passwords.
- pandas to make the SQL outputs into a more readable table.
- GitHub: To guarantee consistent development and high-quality code.
- Pycharm: Our primary integrated development environment (IDE) for writing and debugging code.

Process Achievements:

- User authentication: A system that securely compares passwords with hashed data and verifies user existence has been implemented.
- Unit testing: To guarantee dependability under a range of circumstances, including edge situations, important functionalities.
- Database Integration: SQL queries for user administration were handled effectively.

Challenges:

- Our main challenge has been working very quickly to redesign our product demonstration to a console app rather than a webapp.

Agile Development:

During the development process, we used several Agile components, including: iterative development, code reviews, refactoring and weekly standups.

Implementation Challenges:

- Time restrictions: we are only a group of three, so we all had to do more work.
- Lack of feedback: had we received feedback for our assignment, we may have been signposted to Jinga which would have allowed us to properly prepare for the flask application. This would have better prioritised our planning and learning time and we would not have needed to switch to a console app to meet the deadline.

**See appendix 5 for examples of our team work on GitHub and Trello.*

Testing and Evaluation

Examining and testing

Testing Strategy

Unit testing and integration testing are the two main focusses of the system's testing strategy.

Unit testing: We make sure that discrete actions, like user-related operations and authentication, perform as anticipated. To ensure code accuracy and reliability, test cases for diverse scenarios are written using the unittest framework.

Sample scenarios for testing:

- confirming the presence of a user in the database.
- Verifying the password's consistency following encryption.
- Ensuring accurate entry of new user information into the database.
- Integration Testing: To make sure that several components operate together without a hitch, integration tests will be included.

This involves verifying that authentication works as planned when connected with the database and with front-end form submissions, as well as evaluating the connection with the database itself.

Tools Employed

For composing unit tests, use unittest.

GitHub: To guarantee consistent development and high-quality code, use it for version control, teamwork, and code reviews.

Pycharm: Our primary integrated development environment (IDE) for writing and debugging code.

Functional and User Testing

Functional testing verifies that the system operates as intended when completing operations like password verification, user registration, and authentication. Key user stories are covered by these tests.

User registration involves confirming that a new user can successfully register and that the database has their information (password, email address, and username, for example).

User Testing:

- User testing.
- Login flow.
- Error Handling.

System Limitations

The system has certain drawbacks, even though the current implementation of basic authentication and user management functions effectively:

Password Security: Because of MD5's flaws, the system presently stores passwords using MD5 hashing, which is deemed unsafe for usage in contemporary applications. A more secure hashing algorithm, like bcrypt or argon2 can be used in place.

Scalability: The system may not be optimised for large-scale deployments or high traffic because it is meant for small-scale use. Optimisations like load balancing, database sharding, or employing cache layers (like Redis) could be required when the user base expands to manage more requests effectively.

Locking a database: Now, the system runs a database query for each attempt at authentication. Under high demand, this method might not scale well and could cause locking problems or performance bottlenecks. Performance may be increased by using optimised database queries or connection pooling.

Existing tests simulate database interactions using a live database. While this is helpful for unit testing, to make sure the system functions correctly with the real data sources, end-to-end testing using a live database is necessary.

Structure for Webapp (development)

To make sure the app had a simple and beautiful user experience, we produced three HTML templates and one CSS stylesheet:

Three templates in HTML:

- Landing/Login page (index.html)
- The dashboard page.
- Trips page.
- JavaScript: Since we just needed a few JS functions to show and hide form components, we didn't need to utilize a separate JavaScript file. The HTML templates themselves were used to write them.

Stylesheet for CSS:

- All of the pages' common styling, including the background picture and standardized design elements, were taken care of by the CSS.
- Background Image: The image was chosen to match the app's purpose of creating a calm vibe with components like a sky and palm plants.
- Common Containers: By using containers, all pages were able to have a unified appearance and feel.
- Form and Table Styling: To maintain visual consistency, all input forms and tables were given a unified style.

UAT Test Scrip

We also designed the test script for the HTML scripts. This would have been implemented for the flask application.

**See appendix 6 for a copy of the UAT test script.*

Conclusion

We set out to create a webapp which would allow users to create accounts, input their trips and group travel costs, and update us with their contributions.

Although we had to pivot to a webapp for the deadline, this project has been an enjoyable learning curve for the three of us. The challenges we faced, and the way in which we navigated these changes, have taught us a lot of adaptability and resilience for entering real work projects. We have each worked very hard and pushed ourselves well out of our comfort zones to learn more skills and contribute to the group efforts.

The console app demonstrates a great proof of concept for our coding designs and flows. We are very proud of the work we have produced, and we feel very encouraged by the possibility of further developing this code into a webapp with the HTML demo we have also provided.

Thank you.

Appendix 1: Group Activity Log

Name: Chahat **Project Group (Name/Number):2 (IST)**

This activity log is for you to document your individual contributions and ideas you have done towards your final group project. This will be submitted to your instruct code. You can document how many minutes you spent working on your code, when you had team meetings outside of the class sessions, or use this as a way to viewing. This is to be completed individually.

Week:	Date:	Activity/Task:	Time spent on Activity:	mbers Involved (if ap	Completed
1	24/07/2024	*Reviewed Group Project Kick Off Meeting Recording*	30 mins	All team members	Yes
1	25/07/2024	Read through the Assignment and Marking criteria	10 mins	1 person	Yes
1	26/07/2024	Completed my SWOT analysis	10 mins	1 person	Yes
1	26/07/2024	Friday Group Call	2 hours	All team members	Yes
2	27/07/2024	Answered some of the prompts and handed over to Clarisse	1 hour	2 Persons	Yes
2	02/08/2024	Friday Group Call	2 hours	All team members	Yes
2	03/08/2024	Started working on SQL Databases and Tables	2 hours	1 person	Yes
3	08/08/2024	Group Call in class	30 mins	All team members	Yes
3	09/08/2024	Friday Group Call	1 hour	All team members	Yes
3	09/08/2024	Created and showed the database to all the team members	1 hour	1 person	Yes
4	12/08/2024	Review of Clarisse's purposed new code structure.	30 mins	1 person	Yes
4	12/08/2024	Group Call before class. All talked about new design.Work allocated to everyone	45 mins	All team members	Yes
4	12/08/2024	Researched how to output the html inputs	30 mins	1 person	No
5	20/08/2024	Watched videos on testing the python code	1 hour	1 person	Yes
5	21/08/2024	Learned how to create unit test cases	2 hours	1 person	Yes
5	23/08/2024	Had a group call with sophia	30 mins	All team members	Yes
5	23/08/2024	Call with clarisse to understand the pivot change to console app	2 hours	2 Persons	Yes
5	23/08/2024	Coded the unit test cases with respect to python code	3 Hours	1 Person	Yes
5	24/08/2024	Sent the unit test cases to clarisse to run at her end	30 mins	1 Person	Yes
5	24/08/2024	Submitted the write up for the documentation of our group project	1 Hour	1 Person	Yes

Name: Clarisse **Project Group (Name/Number): 2 (IST)**

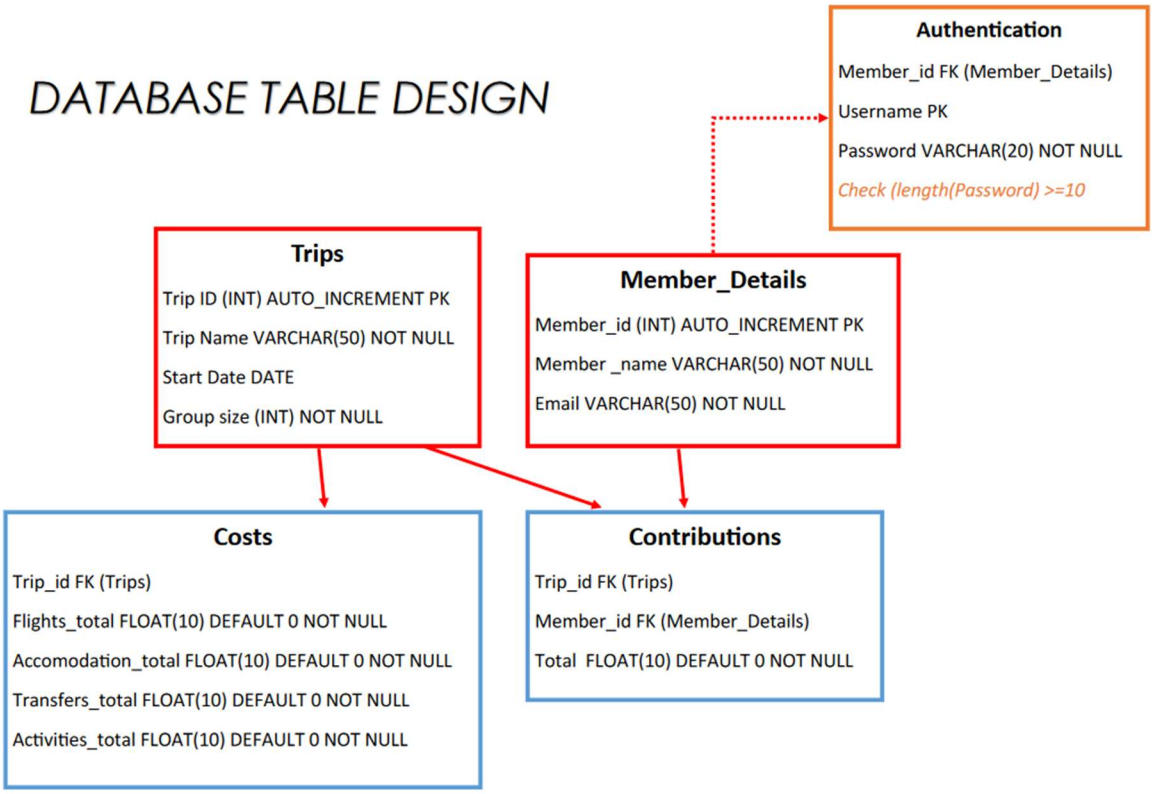
This activity log is for you to document your individual contributions and ideas you have done towards your final group project. This will be submitted to your instructors along with your project code. You can document how many minutes you class sessions, or use this as a way to take notes for your own viewing. This is to be completed individually.

Week:	Date:	Activity/Task:	Time spent on Activity:	Team Members involved (if applicable)	Complet
1	24/07/2024	Reviewed Group Project Kick Off Meeting Recording	30 minutes	All Team Members	Yes
1	24/07/2024	Create Github repository, creat readme, write up minutes, find and share HTLM/Flask/Python link	1.5 hour	1 Person	Yes
1	25/07/2024	Read through the Assignment notes and Marking Criteria	20 mins	1 Person	Yes
1	26/07/2024	Completed my individual SWOT analysis and uploaded it to Trello.	10 mins	1 Person	Yes
1	26/07/2024	Friday Group Call	2 hour	All Team Members	Yes
1	26/07/2024	Caught up minutes, pushed to github, made SSH key for repository, moved activity log to google sheets.	45 mins	1 Person	Yes
2	29/07/2024	Made handout for group assignment. Combined Chahat's work with the prompts I needed to answer.	2 hours	2 Persons	Yes
2	02/08/2024	Friday Group Call	2 hours	All Team Members	Yes
2	02/08/2024	Finalised the Handout and email to Jessie	5 minutes	1 person	Yes
2	02/08/2024	Updated minutes, pushed to github, added assignment handout to github, created gitignore, updated ReadMe file.	1 hour	1 person	Yes
3	07/08/2024	Watched video Jessie sent on adding HTML templates to python.	20 mins	1 person	Yes
3	08/08/2024	Group call in class, Jessie demo	30 mis	All Team Members	Yes
3	09/08/2024	Looking through workflow design, planning python codes, adding notes to one note.	20 mins	1 person	Yes
3	09/08/2024	Friday Group Call	45 mins	All Team Members	Yes
3	11/08/2024	Normalise SQL database, get tables working, insert 'dud' mock value.	1 hour	1 Person	Yes
3	11/08/2024	Map out relationship between the front-end and the back-end to aid building app.	2 hours	1 Person	Yes
3	11/08/2024	Start writing app.py etc	1 Hour	1 Person	No
4	12/08/2024	Group call before class. Demoed SQL. All talked through new design. Work allocated for the week.	45 mins	All Team Members	Yes
4	12/08/2024	Pushed audit note updates to main, amended visuals, SQL branch pull request made, updated relevant notekeeping.	20 mins	1 Person	Yes
4	15/08/2024	Made authentication table, sorted out encryption. Set up connection between python and database. updated requirements.	1 hour	1 Person	Yes
5	21/08/2024	Watched videos on making webapps, testing inputs, designed python files and classes etc.	3 hours	1 Person	No
5	21/08/2024	Group call, Jessie demoed code, I went through python layout, planned remaining work.	1 hour	All Team Members	No
5	21/08/2024	Trying to understand why flask connection not working to take in HTML variables.	4 hours	1 Person	No
5	22/08/2024	Writing all the python to sql functions for the index.html page in console.	3 hours	1 Person	Yes
5	22/08/2024	Group meeting, plan of action, trying to debug html request	30 mins	All Team Members	No
5	22/08/2024	Project admin: updating and pushing minutes, updating readme so it's near finished and easy for Sophia to follow.	1 hour	1 person	Yes
5	22/08/2024	Writing all the python to sql functions for the dashboard page in console.	4 hours	1 person	Yes
5	23/08/2024	Finished the coding python-sql scripts/classes.	1 hour	1 person	Yes
5	23/08/2024	Group meeting (with Sophia), plan to pivot approved. Confirmation that API counts in development branch, dicussed how to organi	1 hour	All Team Members	Yes
5	23/08/2024	Meeting with Chahat, code design and demo.	1 hour	Clarisse and Chahat	Yes
5	23/08/2024	Sorting out the github repo, merging/pulling branches etc, cleaning up the branches.	1 hour	1 person	Yes
5	23/08/2024	Designing the console code logic to mirror flask. start writing code.	3 hours	1 person	Yes
5	23/08/2024	Create visual aid for the python logic, create documentation skeleton file, write up documentation.	2 hours	1 person	Yes
5	24/08/2024	Finish coding the index console substitute, code all the console dashboard substitutes.	6 hours	1 person	Yes
5	24/08/2024	Debug Testing scripts Chahat sent. Merge testing into python-console branch, deleted testing branch.	1 hour	1 person	Yes
5	25/08/2024	Finish coding all console. Debug console test run problems.	4 hours	1 person	Yes
5	25/08/2024	Combine documentation, tidy up work tree, update readme, submit to Sophia.	1 hours	All Team Members	Yes

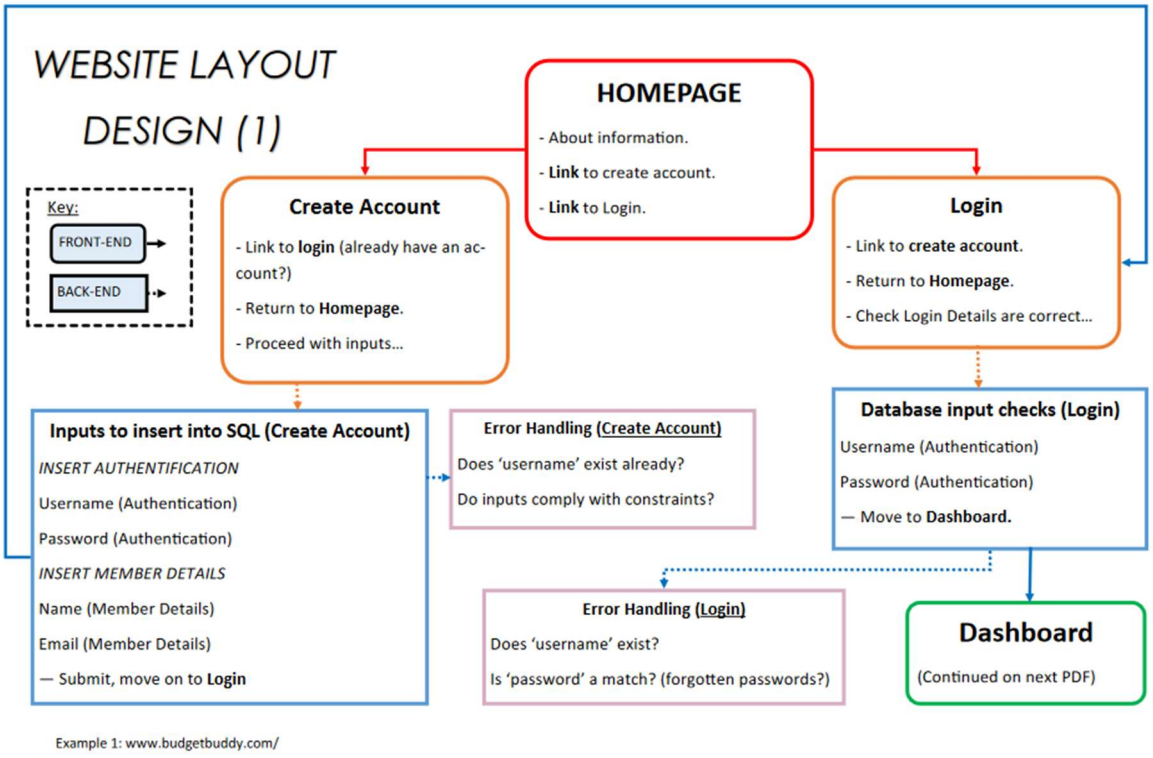
S

Name:	Jessie	Project Group (Name/Number):				
ou have done towards your final group project. This will be submitted to your instructors along with your project code. You can document how many minutes you spent working on your code, when you had team meetings outside of the class sessions, or use this a						
Week:	Date:	Activity/Task:	spent on Act	ers Involved (I	Completed?	Notes:
1	24/07/24	Reviewed Group Project Kick Off Meeting Recording	30 minutes	1 Person	Yes	
1	24/07/24	Read through the Assignment notes and Marking Criteria	30 minutes	1 Person	Yes	
1	24/07/24	Watched the video on HTML and Python Linking, shared by Clarisse	20 mins	1 Person	Yes	
1	24/07/24	Created Trello board and invited the other group members to join.	40 mins	1 Person	Yes	
1	24/07/24	Updated and share the OneNote Notebook we used in the kick-off meeting. Shared with group members.	30 mins	1 Person	Yes	
1	24/07/24	Completed my individual SWOT analysis and uploaded it to Trello.	30 mins	1 Person	Yes	
1	26/07/24	Friday Group Call	2 hours	All Team Mem	Yes	
2	31/07/24	Worked on updating the data mapping diagram, per feedback from Clarisse.	30 mins	1 Person	Yes	
2	02/08/24	Friday Group Call	2 hours	All Team Mem	Yes	
2	02/08/24	Finalized the data mapping diagram, per discussions on the call	30 Mins	1 Person	Yes	
2	02/08/24	Merged pdfs for Assignment submission and submitted to Slack.	15 Mins	1 Person	Yes	
2	04/08/24	Worked on creating HTML and CSS for the Homepage.	5 hours	1 Person	No	
3	05/08/24	Reconfigured HTML and CSS folder/file structure	1 hour	1 Person	Yes	
3	05/08/24	Created new branch in Github for HTML and cloned project to my local pycharm repository	15 mins	1 Person	Yes	
3	05/08/24	Updating of documentation	15 mins	1 Person	Yes	
3	06/08/24	Added HTML and CSS branches to Github and pushed to the Remote Repository	1 hour	1 Person	Yes	
3	08/08/24	Created the html file for the New Trip data entry page and updated the CSS file with relevant classes. Pushed to github.	2 hours	1 Person	Yes	
3	08/08/24	Group call in class, gave demo of the html webpage navigation	30 mis	All Team Mem	Yes	
3	09/08/24	Friday Group Call	45 mins	All Team Mem	Yes	
3	11/08/24	Worked through some alternative SQL structure/table/queries, to see how it might work, given Chahat is running into issues.	5 hours	1 Person	Yes	Not used, went with Clarisse's revamp instead.
4	12/08/24	Review of Clarisse's proposed new code structure.	30 mins	1 Person	Yes	
4	12/08/24	Group call before class. Demoed SQL. All talked through new design. Work allocated for the week.	45 mins	All Team Mem	Yes	
4	12/08/24	Created Butter account, for recording future meetings, as Chahat and Clarisse have used up their credits.	30 mins	1 Person	Yes	
4	12/08/24	Renamed the HTML files and added the hidden "Create New Trip" form. Screenshots shared on slack group discussion for review.	3 hours	1 Person	Yes	
4	13/08/24	Conversation via group slack channel, re the scope of the project and Member Details inputs	30 mins	All Team Mem	Yes	
4	14/08/24	Updates to the HTML and CSS. Documenting the input names for the Python variables on the OneNote workbook	2 hours	1 Person	Yes	
4	16/08/24	Added Login and Create New Account forms to the index.html file. Added "Logout" navigation back to the index page, from Dashboard.html	3 hours	1 Person	Yes	
4	16/08/24	Updated Trello board with HTML progress and other documentation.	20 mins	1 Person	Yes	
4	18/08/24	HTML/CSS: Updated the hide/display function of the forms. Cleaned up unused code and pushed all the recent changes to Github.	90 mins	1 Person	Yes	
4	18/08/24	Researched how to output SQL data from Flask to a HTML table.	30 mins	1 Person	Yes	
5	20/08/24	Added comment notation to all the HTML pages.	30 mins	1 Person	Yes	
5	20/08/24	Added an output table to the dashboard page, to display a summary of all saved trips, from the sqlpython queries, including nav hyperlinks.	2 hours	1 Person	Yes	
5	21/08/24	Group call, Jessie demoed code, I went through python layout, planned remaining work.	1 hour	All Team Mem	No	Crunchtime.
5	21/08/24	HTML/CSS: Added the output table to the trip page. Updated stylesheet with hover over colouring and centered the text in data cells.	2 hour	1 Person	Yes	
5	21/08/24	Made a start on the UAT Testing script for Chahat to test my HTML coding.	30 mins	1 Person	No	
5	22/08/24	Group meeting, plan of action, trying to debug html request	30 mins	All Team Mem	No	Meeting with Sophia tomorrow.
5	23/08/24	Group meeting (with Sophia), plan to pivot approved. Confirmation that API counts in development branch, dicussed how to organise repo.	1 hour	All Team Mem	Yes	
5	24/08/24	Review merged HTML/CSS in GitHub and found some lines inserted incorrectly. Corrections made directly in Github and commited.	20 mins	1 Person	Yes	
5	24/08/24	Wrote up notes on the HTML/CSS coding process, for the submission documentation.	1 hour	1 Person	Yes	
5	25/08/24	Created HTML Demo video for the submission document and presentation.	45 mins	1 Person	Yes	
5	25/08/24	Finished the UAT scripts that would've been used for the webapp, to include in our document.	90 mins	1 Person	Yes	
5	25/08/24	Proof reading comments and non-active code, for typos.	1 hour	1 Person	Yes	

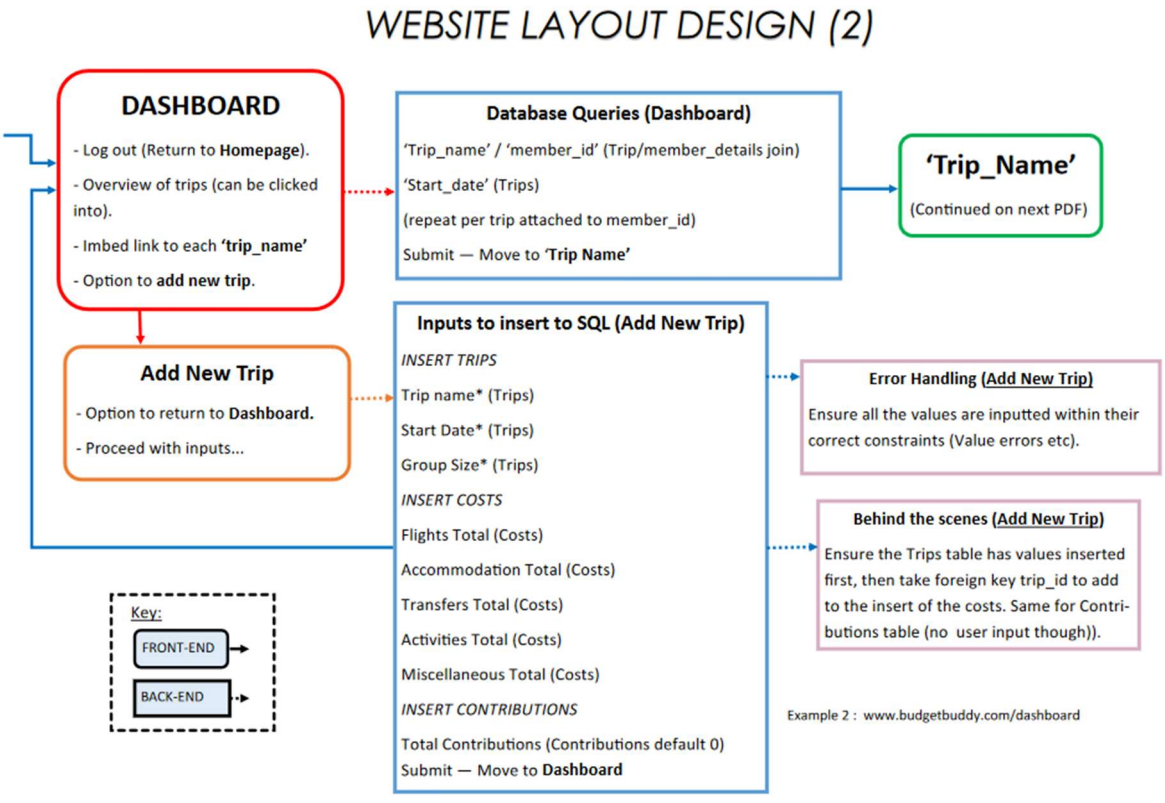
Appendix 2: Database Design



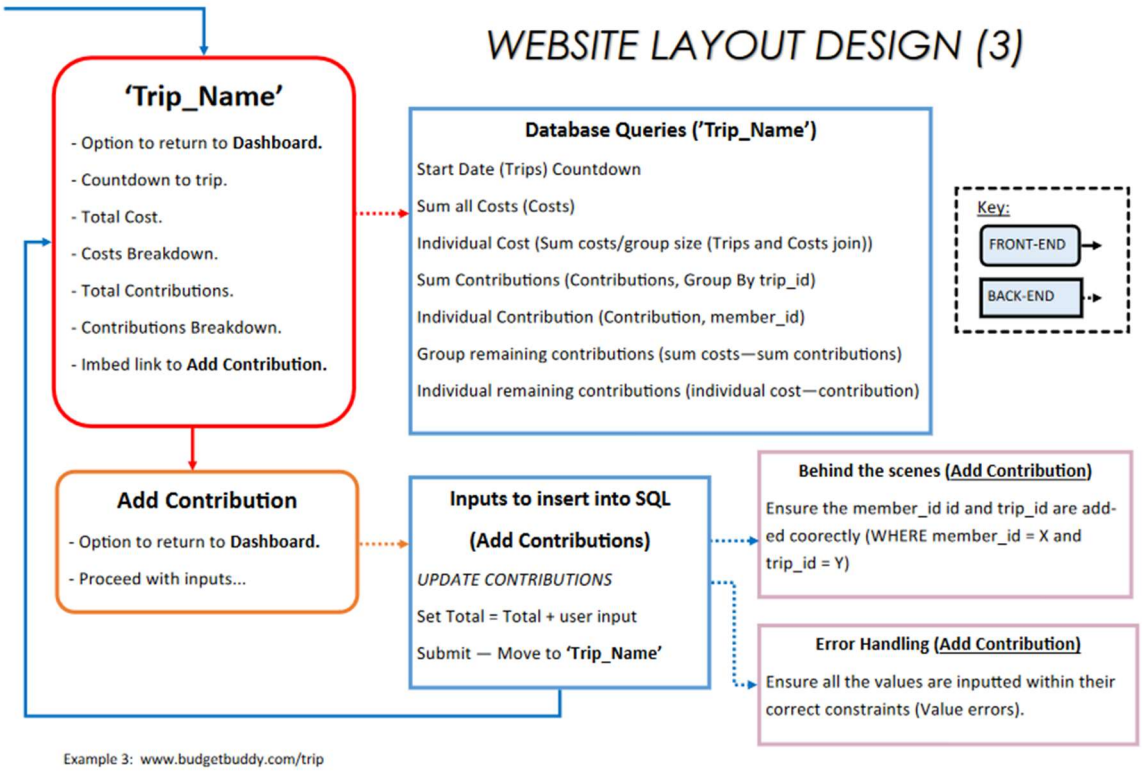
Appendix 3: Webapp Design



Example 1: www.budgetbuddy.com/



Example 2 : www.budgetbuddy.com/dashboard

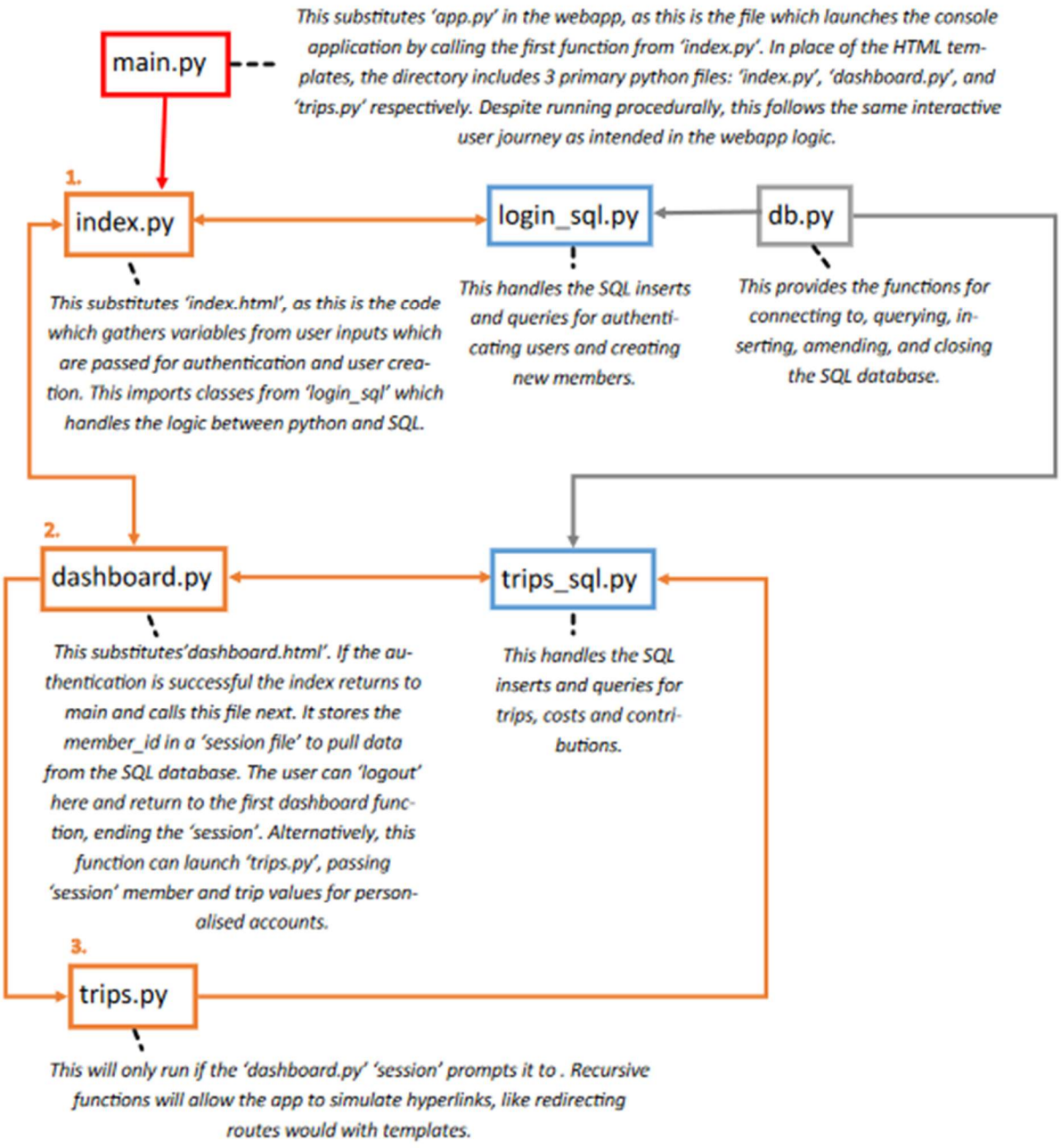


Example 3: www.budgetbuddy.com/trip

Appendix 4: Python Logic

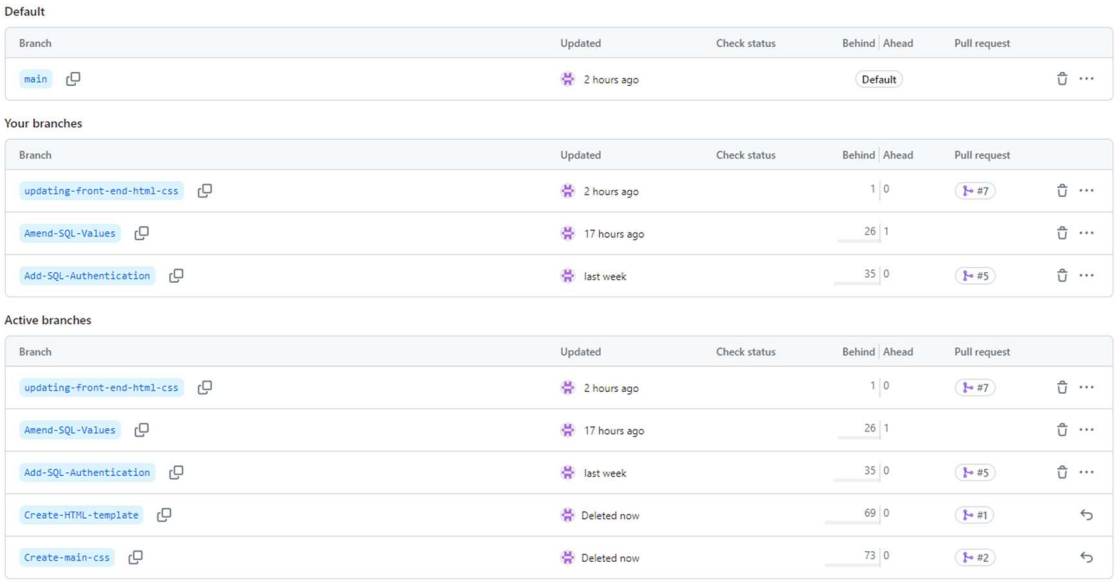
Python Console Logic:

NB: This code implementation has been designed to placeholder a flask webapp. Therefore, the files have been organised in a way which can later be converted back to the HTML templates which have been moved to a development branch due to time constraints.

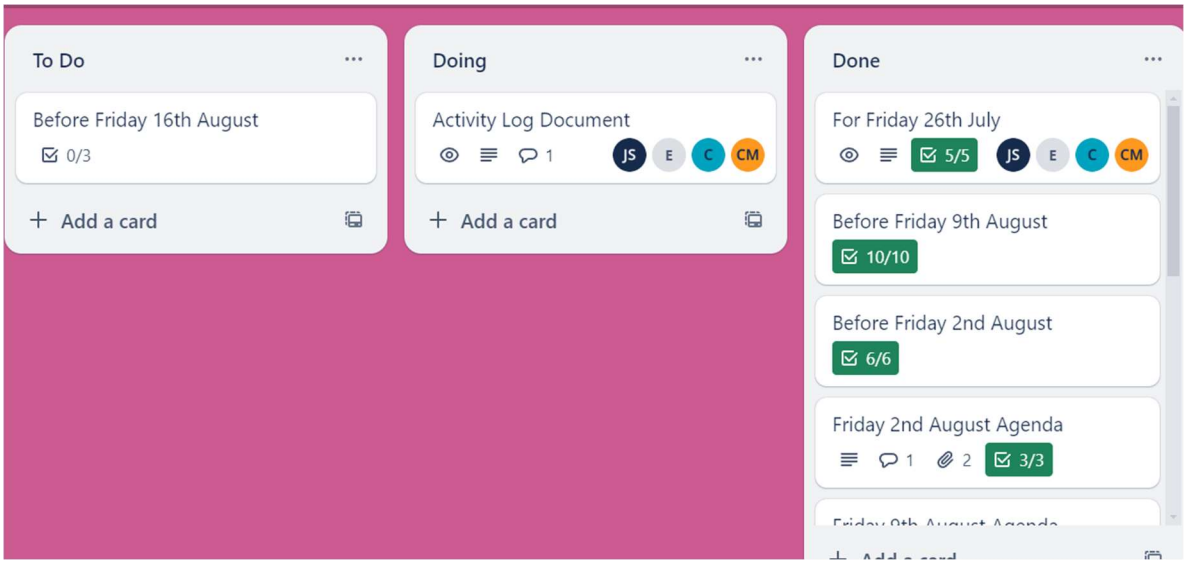


Appendix 5: Examples of Teamwork

Github Branches:



Trello Boards:



Appendix 6: UAT Test Script

TEST
Index/Login Page
Open file in browser from Pycharm
Resize browser window.
Click on Login button
Enter details into the Login Form
Click Submit button
Close browser and reopen the Index.html file
Click the Create Account button
Enter details into the Create Account Form
Enter invalid email address
Enter password
Click Submit button
Enter new credentials into the Login Form and click submit
Dashboard Page
On page load
Click Create New Trip Button
Click Submit
Trips Page
On page load
Enter details into the input fields of the form and click submit
Click the Return to Dashboard link
Dashboard Page
On page load
Hover over the summary table of saved trips
Click on one of the hyperlinks in the Trip Name column
Trips Page
On page load

Expected Outcome
Page loads Background Image covers window Positioning of elements is correct and not skewed.
Background Image resizes and isn't skewed. Elements move dynamically, relative to the window and background image
Login form appears
Expected data type makes sense
Navigates to the Dashboard Page
Create Account form appears
Expected data types make sense
Message appears, prompting the required format for valid email address
Password input is masked on the screen
Create Account form disappears and Login Form appears Message "Please login with your new credentials" appears at the top of the login form
Navigates to the Dashboard Page
Table of previous trips is displayed Hyperlinks to drill down into the details of each individual trip Create a New Trip button displays Create a New Trip Form is hidden Logout option displayed in top right navigation bar
Create New Trip Form opens Expected data types make sense
Navigate to the Trips Page
Trip details table loads, headers only, as this is a new trip Form to enter new member contributions is displayed Return to Dashboard link appears in the navigation bar
Form resets, in order to accept additional contribution entries
Navigates back to the Dashboard Page
Table of previous trips is displayed Hyperlinks to drill down into the details of each individual trip Create a New Trip button displays Create a New Trip Form is hidden Logout option displayed in top right navigation bar
Row background changes colour, to confirm selection
Navigate to the Trips Page
Trip details table loads, with summary of contributions entered previously Form to enter new member contributions is displayed Return to Dashboard link appears in the navigation bar

RESULT	COMMENT
Pass Pass Pass	
Pass Pass	
Pass	
Pass	
Pass	Authentication to be added when the api is connected
Pass	
Pass	
Pass	
Pass	
Pass Pass	
Pass	Authentication to be added when the api is connected
Pass Pass Pass Pass Pass	Hardcoded mock data, to be replaced by api data from the db
Pass Pass	
Pass	
Fail Pass Pass	Hardcoded data appearing, should be fixed when api is connected
Pass	
Pass	
Pass Pass Pass Pass Pass	
Pass	
Pass	
Pass Pass Pass	Same data loading for each trip, will be fixed when api connected