Hi Judges, here will give a short gist of what we did, thanks for the opportunity. We have chosen JavaScript as our theme, we sticked to it as much as we could.

Workflow:

1. Login and registration modules are connected using RESTful API written in loopback framework, the users can register, login. Mandatory fields are ensured for data validation
2. Our landing page is created in HTML, CSS and JavaScript as per theme, all use cases have been successfully completed and demo snaps / PowerPoint are attached.

Mandatory use cases (we have fulfilled these to the best of our knowledge)

1. Add Transaction
2. Remove transaction
3. Edit transaction
4. Show expense
5. Show income
6. Show available balance
7. Display Transaction history

Innovations (these are the features which were not defined in the use case, but we included to enhance the user experience)

1. Chat feature to act like a sticky note
2. Date field for ease of transaction tracking and sorting
3. Analytics of the transactions by visualizing in Pie chart and Bar chart
4. Registration and login - We felt it makes the project whole so included them and connected using a DB
5. Dashboard

Usage of GitHub Copilot

1. Copilot prompting in as comments, automatically generated the required code which can identified throughout the code
2. Project is created using plain JS for frontend and loopback for backend, where Copilot helped in creating effective boilerplate in the first place
3. Copilot provided advanced code completion in multiple use cases as mentioned below
   a. Loopback Model creation (User model, Finance History) – Model creation from scratch is a confusing process, but copilot was able to understand the model contents and suggested to add relations between data
   b. RESTful API to access the data models – lifecycle observers were auto completed effectively
   c. User authentication scenarios such as login, register and logout were mostly completed with the help of Copilot in concise way
   d. Error handling suggestions by copilot were to the mark which helped in covering multiple use cases

     e.   Form validation in frontend was auto completed with Copilot

     f.   It generated fetch API endpoint methods on just providing the function name

     g. Few HTML skeleton were also auto generated

     h. Generated most of the analytics using Copiloy

4. Overall, GitHub Copilot played pivotal role in the skeleton and the framing of this web app, which drastically minimised the time, which is a crucial aspect in this hackathon.

Judging Criteria:

1. Clean and organized code - The code is well organized the minimised LOC's and tried giving max output.
2. Easy-to-understand code - we made sure every module and sub module of the code is well organized as easy to understand; even if a new person other than developers access the code, it's easy to understand.
3. No bugs or errors - we minimised the margin of error as this is to be done in 4 hours we tried our best to minimise errors and bugs.
4. Effective use of GitHub Co-pilot - We only used co-pilot and we made the best out of it, it's like a working with a person who is skilled in all technologies
5. It was able to make us save time, and the results were really awesome, more than 60 percent of the effort and time is saved using co-pilot.
6. The solution we developed is stable, reliable, and user-friendly.

Future Enhancements:

1. We have added a chat feature which is still under dev as the time constraint matters, we plan to integrate chatbot with azure bot services in future.
2. This is planned to be converted as Progressive Web App by improving the UI in an extended way
3. This application is planned to be hosted in Azure WebApp that can be accessed by anyone
4. We also plan to display the history of transactions in analytics and use power BI with azure in future

What makes budget buddy standout from other brands.

1. We have a In-app chatbot to guide user on spends
2. We tried to make UI/UX more effective and user friendly than other apps.
3. We plan to add Auth0 for seamless SSO integration and authentication

NOTE:

This is a Minimal Viable Product (MVP) and does not represent the final product. We tried to implement the best in the given time frame