**CSIS-4280**

**Project Title: Pocketbook**

**By:**

**Kosha Raval – 300324116**

**Japneet Singh- 300323547**

Project Title: Pocketbook

Introduction

Budget Management is a crucial aspect when it comes to manage our expenses on a day-to-day basis. In a world where everything is online oriented, generally it becomes tedious to keep track of the paper-based tracks and receipts. People want everything to be automated, with minimal to no use of paper-based records. This is where next generation applications come into play. We all need a budget, be it small or large to stay organised. To curb this, a budgeting app helps the user track and manage their finances via Mobile application.

Our primary aim is to make our application less human dependent which will make it easier to manage our money very conveniently. The application focuses primarily on asking the user for its login details. Based on those details, the user gets registered if he/she hasn’t before. Next, once the user logs In, the application will ask you to enter income and expenses and then set your monthly budget based on your inputs. The user can add fixed monthly expenses and set an alarm to pay them. This would be a one-time entry per month.

We will be using MongoDB as our database which will store login information about the user, his/her monthly expenses and income. Lastly, we will be implementing Flash with python to handle server-side requests.

Functionalities

Below are detailed functionalities and scenarios associated with each activity:

1.Login/logout: A user enters their credentials to authenticate. These credentials are verified against records in a database: if valid – access is granted (corresponding user information is fetched), otherwise rejected. This activity also provides a button link for registration. Following this link, a registration activity fires up.

2.Registration: A user can create an account that will be associated with all their data (it is required to fill in credentials and personal information).

3.Dashboard: Following a successful authorization, this activity provides an overview of expenses, current budget, money spent and savings. It also shows the list of today’s and last 7 days’ expenses. Each item in the list is a link for detailed description.

4.Fixed Income & Expense: Shows a list of fixed incomes, expenses and provides an interface to enter or edit income/expense amount along with information associated with it.

5.Profile: Shows user information and allows for editing. Calculates and displays tax amount based on income and expense categories.

Screenshots with explanation:

1. This is our splash screen with a logo for a budget.

A picture containing text, electronics, monitor, display

Description automatically generated

1. Graphical user interface, application, Teams

   Description automatically generatedThis is our landing home page where user will either log in or register if he/she is the new user
2. This is the registration page where the new user registers. This registration will go through the server and finally in the database.

Graphical user interface, application

Description automatically generated

1. This is the dashboard for our app where we will show all budget related information and here. We get the transaction data that is stored on the server and when retrieved, it displays inside today’s proceedings recyclerview.

A screenshot of a computer

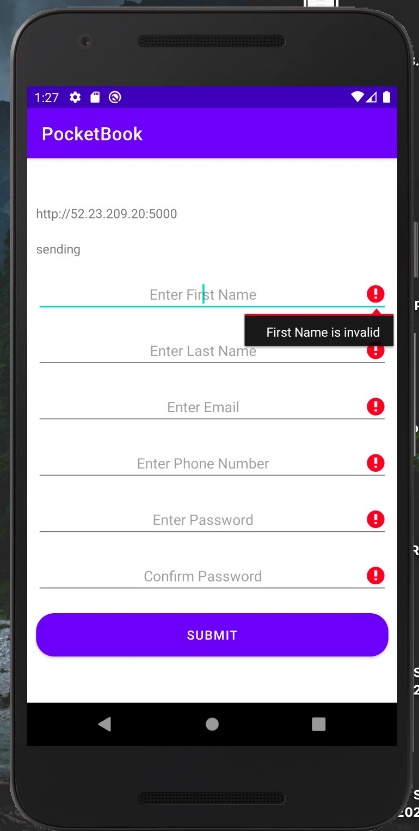
Description automatically generated with medium confidence

1. This is the user profile that is connected to the server and database. We retrieve the information from the database and set the budget for the user.

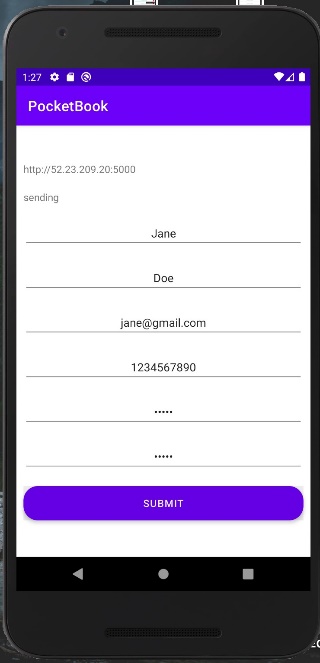
A screenshot of a computer

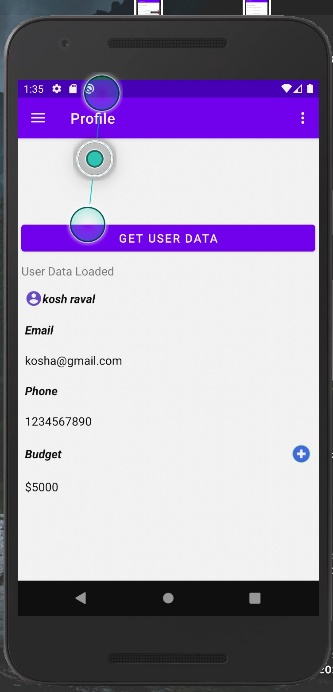
Description automatically generated with low confidence

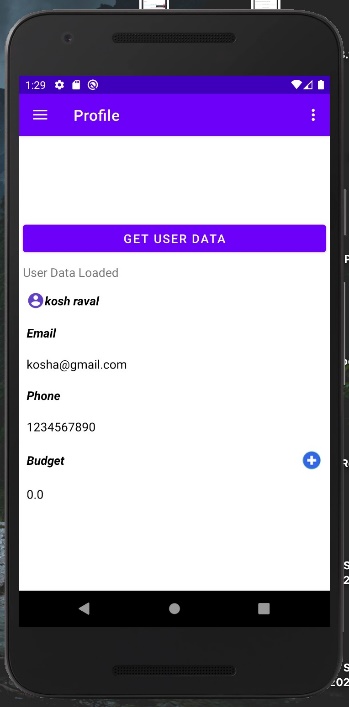
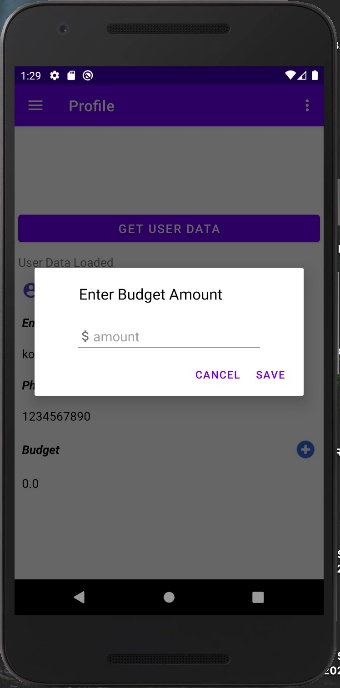
Now, we will enter new user and if its empty, we will prompt the user for details.



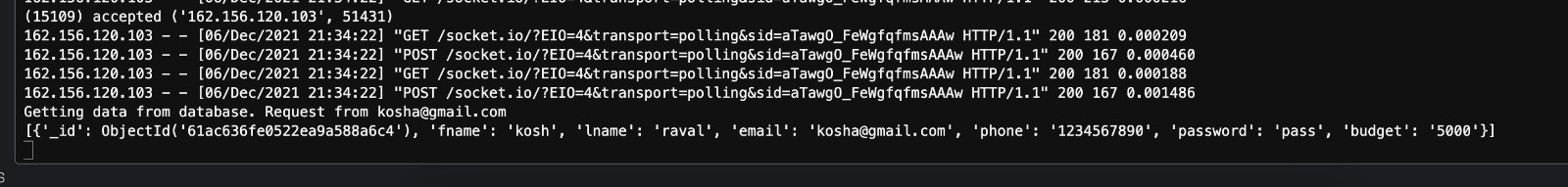
Creating new user jane doe



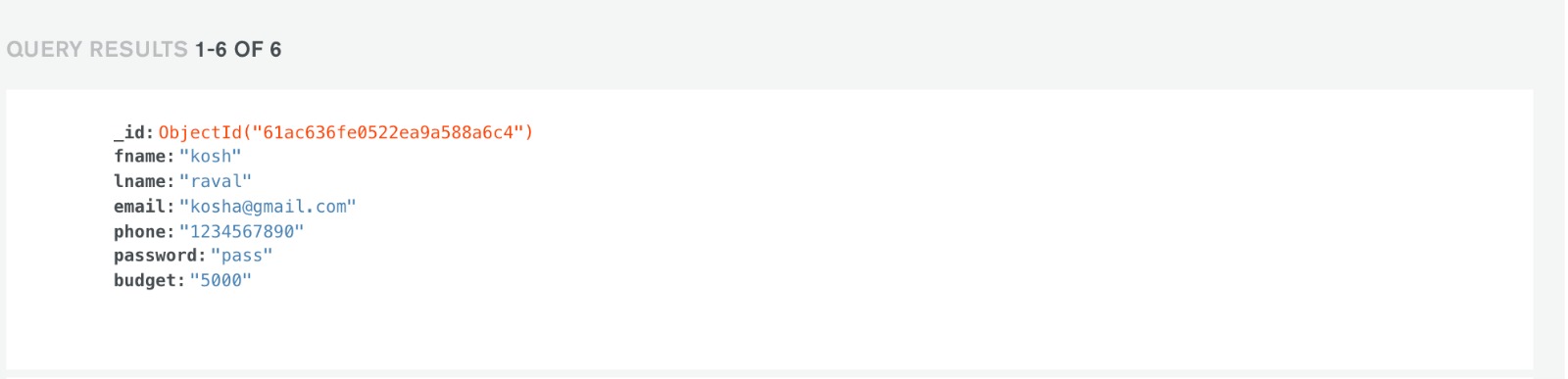
Now we proceed ahead with the profile section. We are here retrieving the user information by clicking the get user data button and it is loading the information from our server and at last from our database.



In our socket.io file, it is getting a request from get user data button we created.



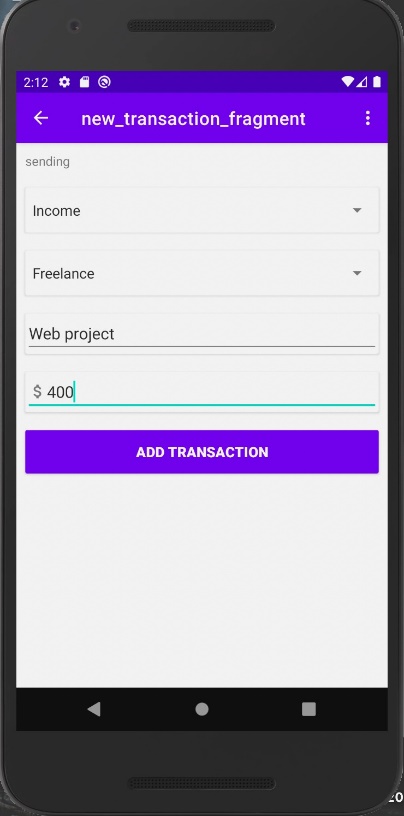
User information getting stored inside database.



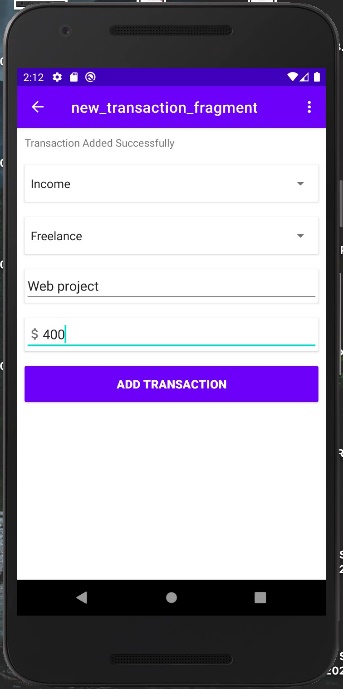
We have implemented choice of incomes, expenses, savings for our user that fits his/her budget needs and can classify.

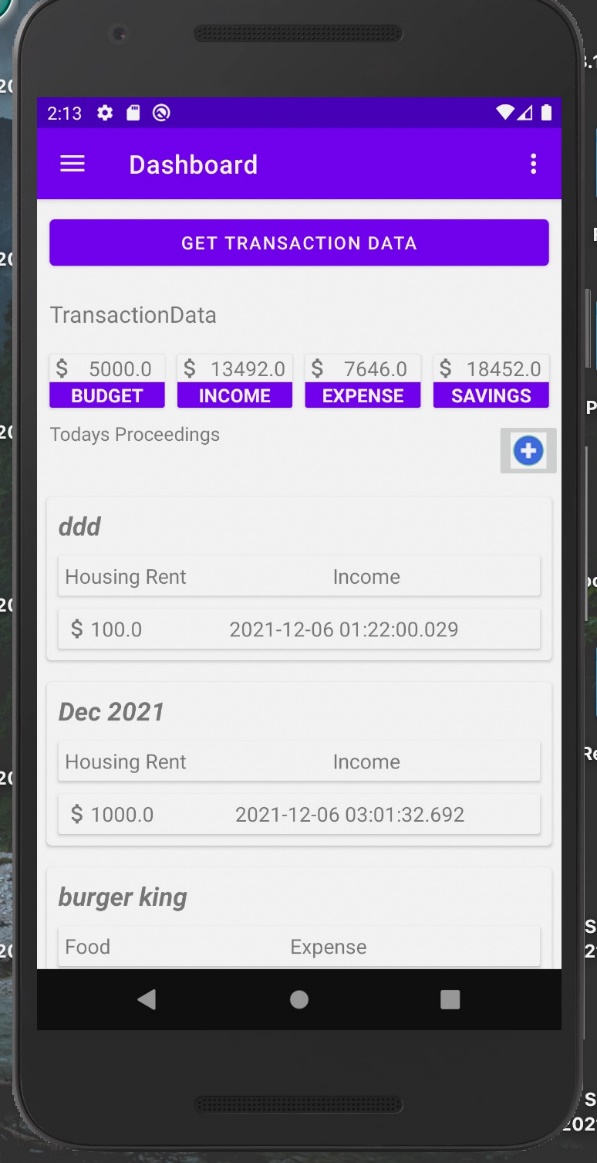
Graphical user interface, application, Teams

Description automatically generated

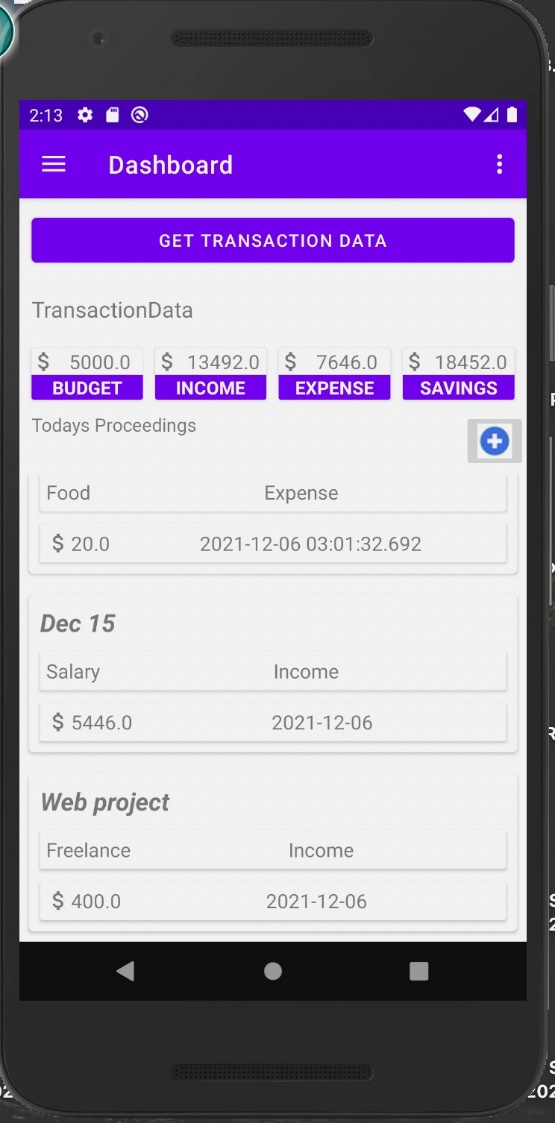
Here, we create a new type of transaction, and it is sending the data to server.

Here, the transaction is added successfully.



Once we return to our dashboard, we will now retrieve the user’s stored information from the server and database.

We are able to successfully add the web project transaction for our user and retrieve its details back to the dashboard.



Here is the reference of our database stored information.

