2020 Year 3 Group Project

Members:

Lochlann O' Neill - R00175741

Keith Bullman - R00178736

Gerardo Jr. Ancheta - R00135146

Daniels Pikurs - R00166279

GitHub Repository:

https://github.com/Lochlannoneill/assignment_groupProject

Table of Contents

Introduction

Project Overview

The global crisis of the Covid-19 pandemic has created widespread disruption. Apart from a crisis in the healthcare sector, there has also been serious economic consequences. Financial institutions, in particular, are exposed to severe economic downturns since a large volume of their business involves lending money which needs to be repaid. Rather than relying on low skilled and poorly paid banking officials to make loan decisions, new technology enables more accurate predictions through the use of AI and appropriate datasets of client information. This transformation in financial services, coupled with new machine learning models and frameworks have created an opportunity for your start-up software development company to develop applications to address a potential tsunami of loan defaults on new business. Your company has been approached by a US bank and tasked with the development of a bespoke prototype system which predicts the likelihood of its customers defaulting on new loans taken out. As part of the negotiations with the company, you have retained the IP and may use it in future projects.

Goals of the Project

- The Android Application allows the user to determine whether or not they are able to take out/repay loans
- The Website, for the broker it let's the broker register and login. The broker is able to enter their details and data in the concept loan. The banker is able to register and login too. Each banker has their own client and the client's data is displayed to each banker's profile.
- The Fin-AI Engine is designed to take in internationally available datasets, and through machine learning algorithms, provide a backend AI that can assist with customers requesting loans and the ability to purchase houses.

Group Roles

Lochlann O Neill - Project Leader, Front-End Mobile Application
Daniels Pikurs - Back-End Mobile Application
Gerardo Jr Ancheta - Website and Database
Keith Bullman - Chatbot and Fin-AI Development

Database Overview

The database consists of numerous tables integrated to the web and mobile application. The database tables include a user profile database for both banker and broker, a loan application data of the broker and a contact us database.

Tables

Table Name	Contents	Link to
broker_db	Broker user profile information. Username, email and password	bankerdata_db
banker_db	Banker user profile information. Username, email and password	bankerdata_db
bankerdata_db	Records of broker information for the Fin-Ai to assess	broker_db banker_db
contact_db	Public client's messages and queries	none

Table Details

Table name: 'broker_db'

Contents: Contain records relating broker's profile information.

Column Name	Contents	Links to	Type
id	Primary Key, auto increments		int(20)
username	Broker 's username	bankerdata_db	varchar(20)
user_email	Broker's email	bankerdata_db	varchar(60)
user_pwd	Broker's password		varchar(255)

Table name: ‘banker_db’

Contents: Contain records relating banker’s profile information.

Column Name	Contents	Links to	Type
id	Primary Key, auto increments		int(20)
admin_username	Banker ‘s username	bankerdata_db	varchar(20)
admin_email	Banker’s email	bankerdata_db	varchar(60)
admin_pwd	Banker’s password		varchar(255)

Table name: ‘bankerdata_db’

Contents: Contain records relating broker’s information in relation to loaning money and some profiling.

Column Name	Contents	Links to	Type
id	Primary Key, auto increments		int(20)
admin_username	Banker ‘s username	banker_db	varchar(20)
broker_username	Broker’s username	broker_db	varchar(20)
broker_email	Broker’s email	broker_db	varchar(60)
broker_age	Broker’s age		int(20)
broker_income	Broker’s income		decimal(20,0)

Website Overview

Website is developed and implemented through HTML, CSS, PHP, MySQL and JavaScript.
The website consists of an index.html which is the home page of the website. In the home page, it contains the links to sections of the home page and links to other pages of the website. The sections of the home page are the about section and contact us section which let's new clients send messages or queries. The website also includes a login page, register page and a register page for the banker (custom URL, a custom register page). Each user has their own dashboard. After the broker login in, they're redirected to their own user dashboard which is a form to input their details in terms of loaning money. For the banker, after login they are redirected to their own user dashboard. For each broker, they have their own banker and in the banker dashboard, it displays the details of the specific broker assigned to them.

Current Features

- Includes a home page, which is broken down into sections. About section and contact us section. Contact us section, public customers are able to send messages or queries.
- Both user's, brokers and bankers are able to login in the same login page.
- Brokers and bankers have their own register page. Bankers have a unique register page, a unique URL.
- Brokers and bankers have their own user dashboards.
- Brokers' dashboard consist of a form for loaning money and their information after filing the form.
- Bankers' dashboard includes a table with datas of their broker's details.
- Terms and privacy page, which contains the terms and privacy of the website and information about terms and privacy. Refer to the GDPR and a reference link to the source documentation of GDPR.

Technologies Used

- Initially, web application development was in WordPress, as sprints progressed, we decided to change to hardcoding the website.
- The Code Editor used for development was sublime and visual studio code.
- I.T. Languages used for implementation scripting were HTML, CSS, JavaScript and PHP.
- For database, MySQL and for database management was phpMyAdmin
- XAMPP was used for cross-platform open-source web servers.

Difficulties Encountered

- A feature within the website, in the user dashboard, a “delete” button within the tables, isn’t functioning. The function was so the user’s are able to delete their information from the database. The scripts are there but the function isn’t working.
- Database queries and scriptings while implementing the website. Intensive research and algorithm mapping created a view and clear understanding of the database queries.

Additional Features from Future Development

- Fin-Ai engine, chatbot and mobile application integration.

Android Application Overview

Current Features

- The user may register for a new account, providing their username, email, etc.
- The user may login to the application using their username and password
- The user may log out of the application
- Side navigation bar provides user-friendly approach at app navigation
- Tabbed layout on the home page provides three fragments for content
- RecyclerView within the tab fragments to provide ability to list details
- Attractive user interface

Technologies Used

- Java programming language used in the development of the application
- XAMPP Control Panel used to control the Apache and MySQL servers
- PhpMyAdmin is used as an open source administration tool for MySQL
- PHP programming language to create scripts for a database connection
- JSON for doing the POST and GET HTTP Calls to the localhost and returning the response in the form of a JSONObject

Difficulties Encountered

- Some links within the menu of the side navigation window (such as ‘support’) will crash the application once clicked. I’m not entirely sure why this is the case considering the code for these activities are practically the same as the activities that do work.
- Change of activity from the Login/Registration activity would switch regardless if the process was successful or not.

Additional Features from Future Development

- The chatbot AI will be incorporated within the mobile application, possibly under the ‘support’ activity.

- The Details tab will be used to incorporate the two different dashboards either for Broker or Banker.
- The Prediction tab will incorporate the predictions for your liabilities.
- The Payments tab will display your financial payment history.
- The RecyclerView within the tab fragments will be used to display a list of items from the database
- In the Register activity, the user will have the option to sign in via google or other options
- When signed in, the side navigation menu will show the details of the user that is currently signed in

Fin-AI Engine & Chatbot Overview

Current Features

- Linking Fin-AI Engine to the client Fin-AI app, and Fin-AI Web application with an API.
- A model, using a publicly available dataset, which determines loan suitability of a given client's financial profile, as well as determining the type of house a client can purchase.
- Reporting which displays the accuracy of the loan suitability and house type models.
- A chatbot which provides automated customer support on the Fin-AI Web.

Technologies Used

- Python 3.6, with various packages such as Keras, Tensorflow, Flask, Pandas, Numpy, sklearn, and NLTK.
- Brief HTML page to render once the chatbot application is exited.

Difficulties Encountered

- Linking AI with Web and Phone applications.
- Connecting to database due to encryption; instead pulls login data from decrypted database CSV file.
- Chatbot does not load complete 'chatbot.html' when chatbot GUI is exited.

Additional Features from Future Development

- Chatbot would render when 'Chatbot' on index.html is clicked, similar to other mainstream customer support automated chatbots. (Eg: Amazon)

Pre-Sprint Preparation

Used a shared GitHub repository to regularly update with our completed work.

Decided on ‘Discord’ being our primary communication platform. Text and voice channels created within the Discord server to accommodate our communication requirements.

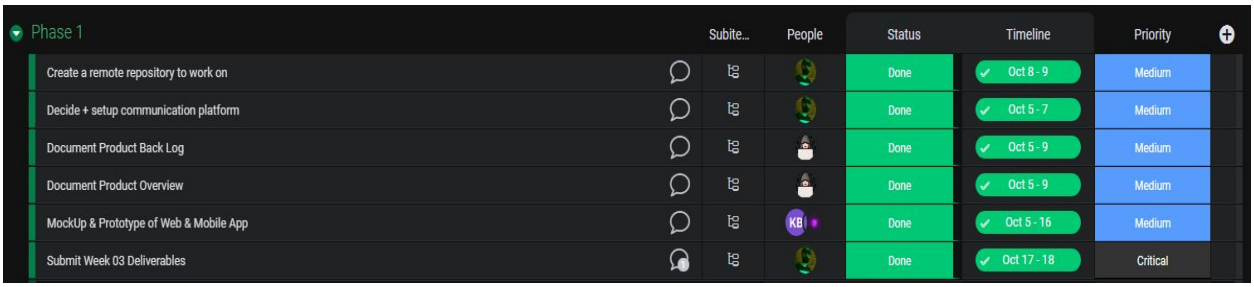
Creation of the product backlog and product overview.

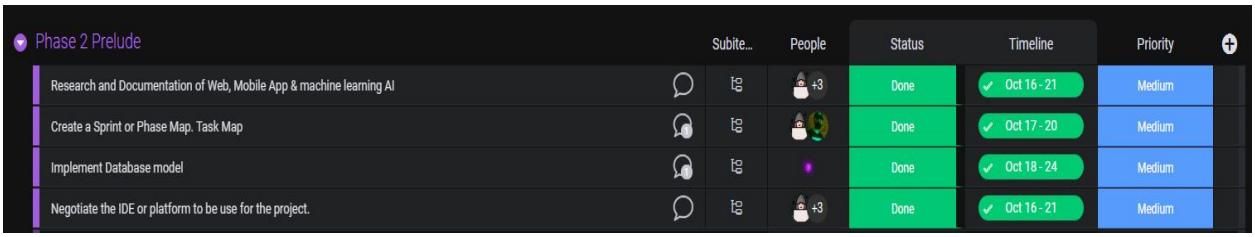
Designed wireframes of the website and mobile application.

Research and documentation of web, mobile & machine learning AI

Implementation of the Database model.

Setup of ‘Monday.com’ to develop sprint map and to provide task assignment to group members





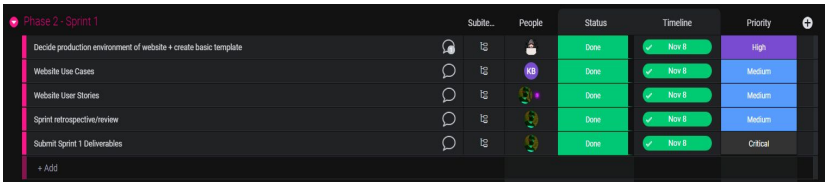
Sprints

Sprint 1 (Oct 16 - Nov 8)

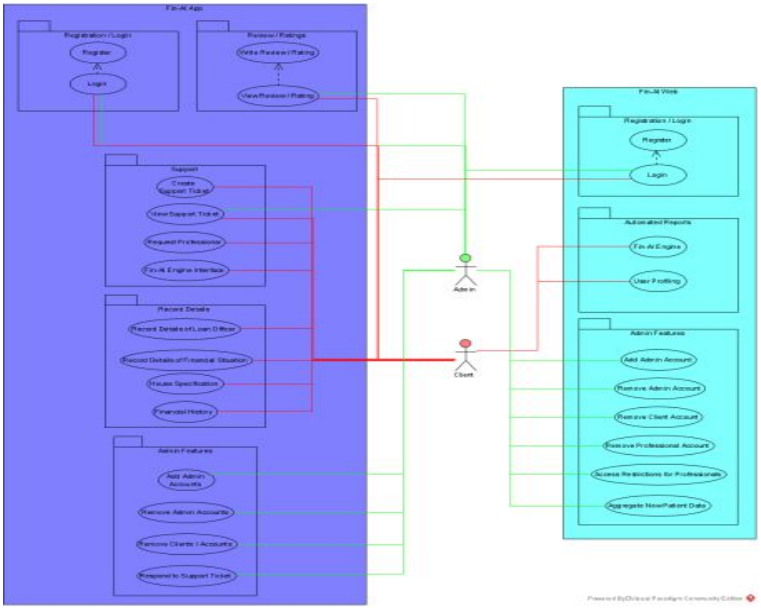
Sprint Backlog

- Creation use cases diagrams and user stories for website and app
- Development of basic template for website and app
- Research into AI for chatbot
- UI Design (Website/App/Chatbot)
- Wireframe + Use Cases

Sprint Details

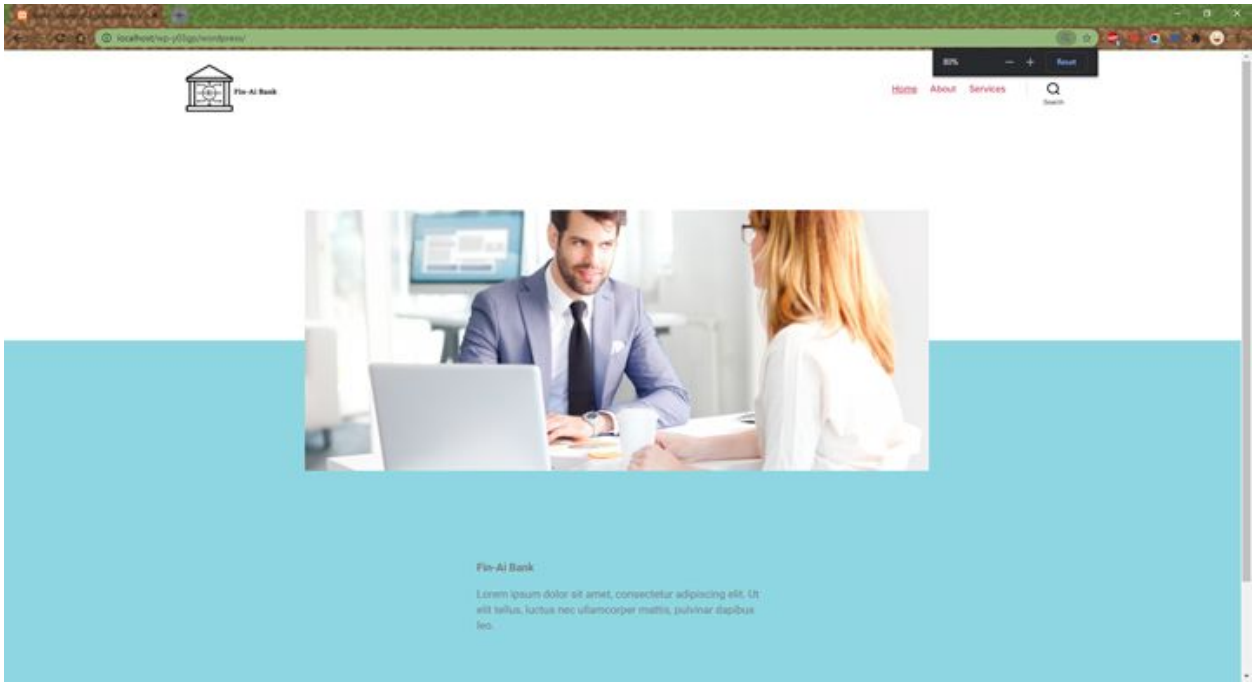


The use of a ‘Monday.com’ board proved to be a much more efficient project management tool for our group than that of a Trello board. It is easy to assign certain tasks and their due dates to each team member. Unlike with a Trello board, all important information is already displayed in a much more simpler and convenient manner.



The Use Cases for both the web application and the mobile application. In the use case there is an ‘admin’ and a ‘client’. This would be changed to ‘banker’ and ‘broker’ in later development.

Original wireframes of the mobile application before actual development. The sign in/register page remained roughly the same, whereas there was a complete overhaul to the rest of the application's design.



Initial development on the website completed. This homepage would just serve as a template to improve on for future progress.

Sprint 2 (Nov 8 - Nov 22)

Sprint Backlog

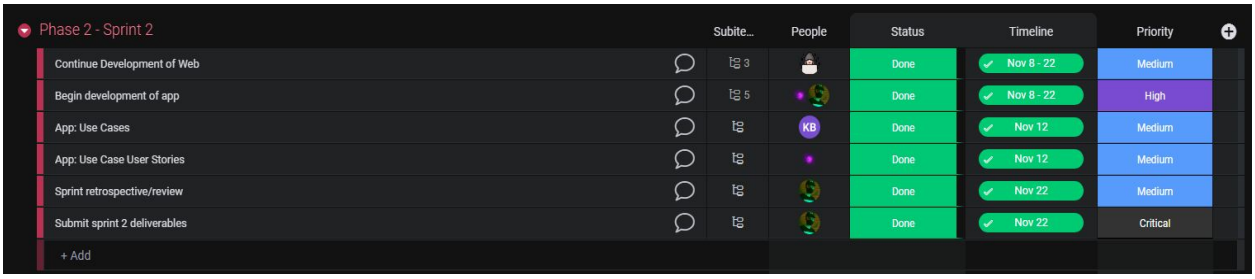
- Banker database table
- Broker database table
- Banker data database table
- “Contact Us” database table
- Website home page
- Website contact section (home page)
- Website registration - client (broker)
- Website registration - banker
- Website login
- Website user profiling
- Website Broker Dashboard
- Website Banker Dashboard
- None user data
- Mobile Side Navigation Window
- Mobile Sign in
- Mobile Register
- Modify database

Sprint Details

This sprint required us to set up all the tables where we would push data to, such as registering users, and pulling data from, for tasks such as signing in clients who exist in the table.

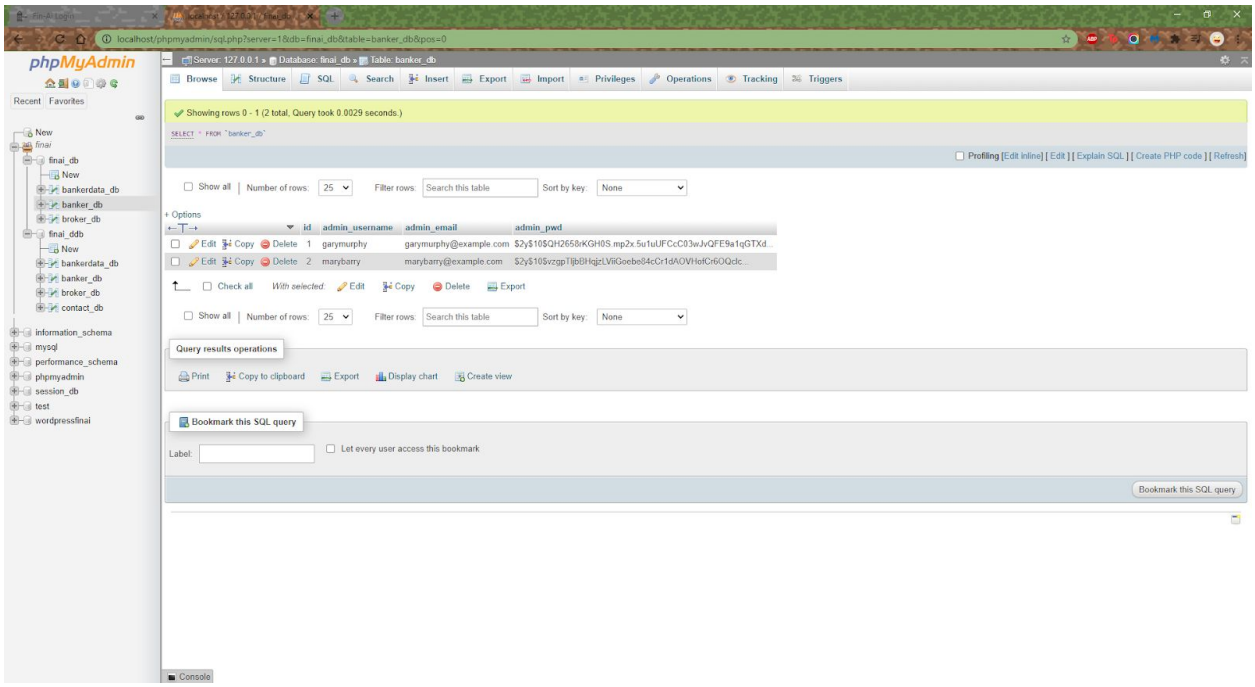
As well as this, we would need to begin setting up the templates for our website pages, and set up pages that will help with certain use cases, such as registering clients or bankers.

Also, we would need to begin the Android application too, and implement use cases/user stories into the application.

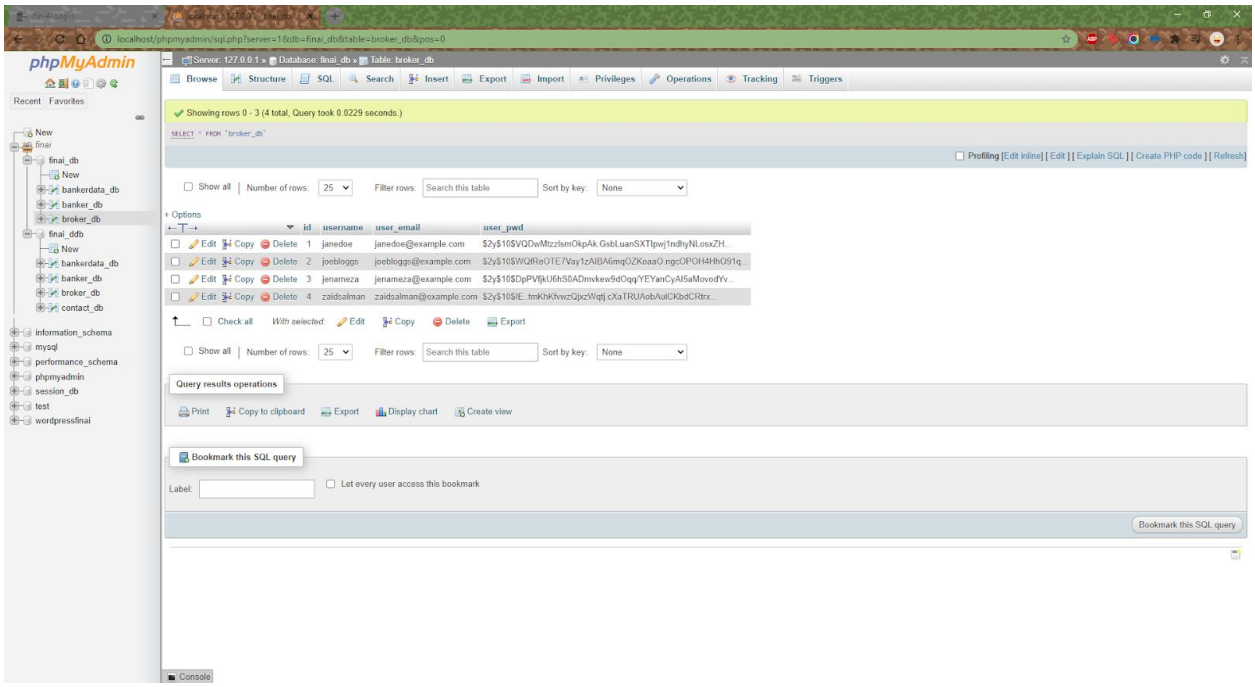


Database

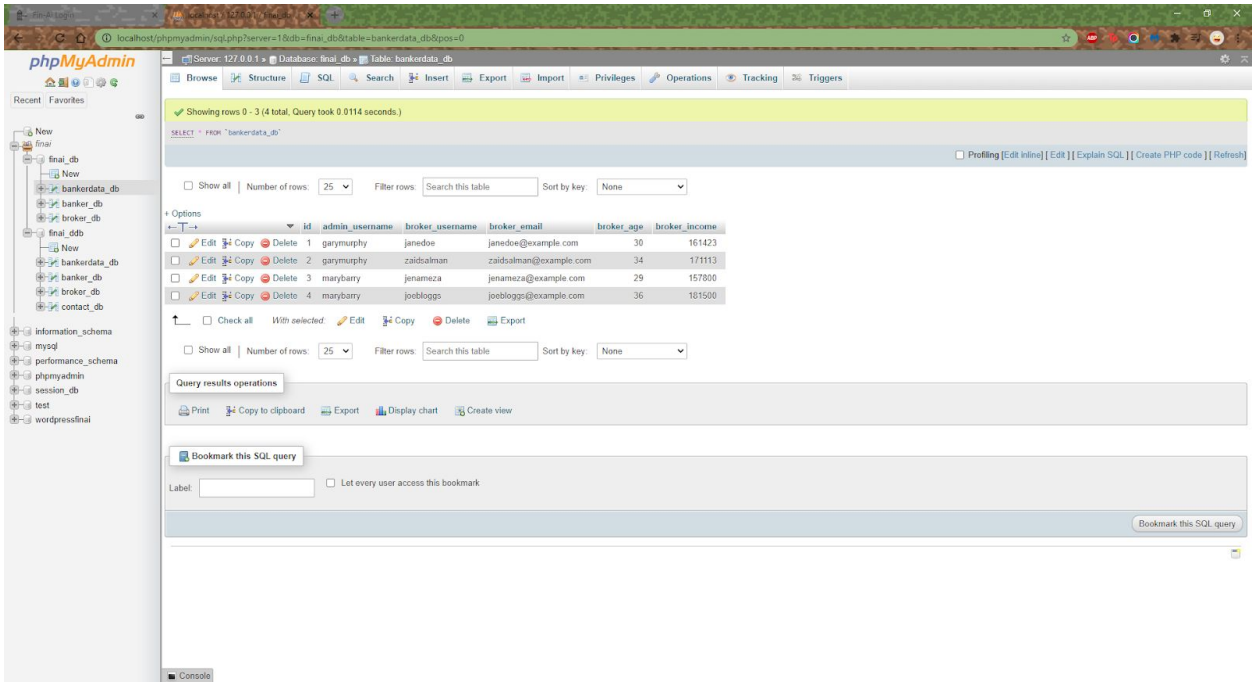
Banker Database table



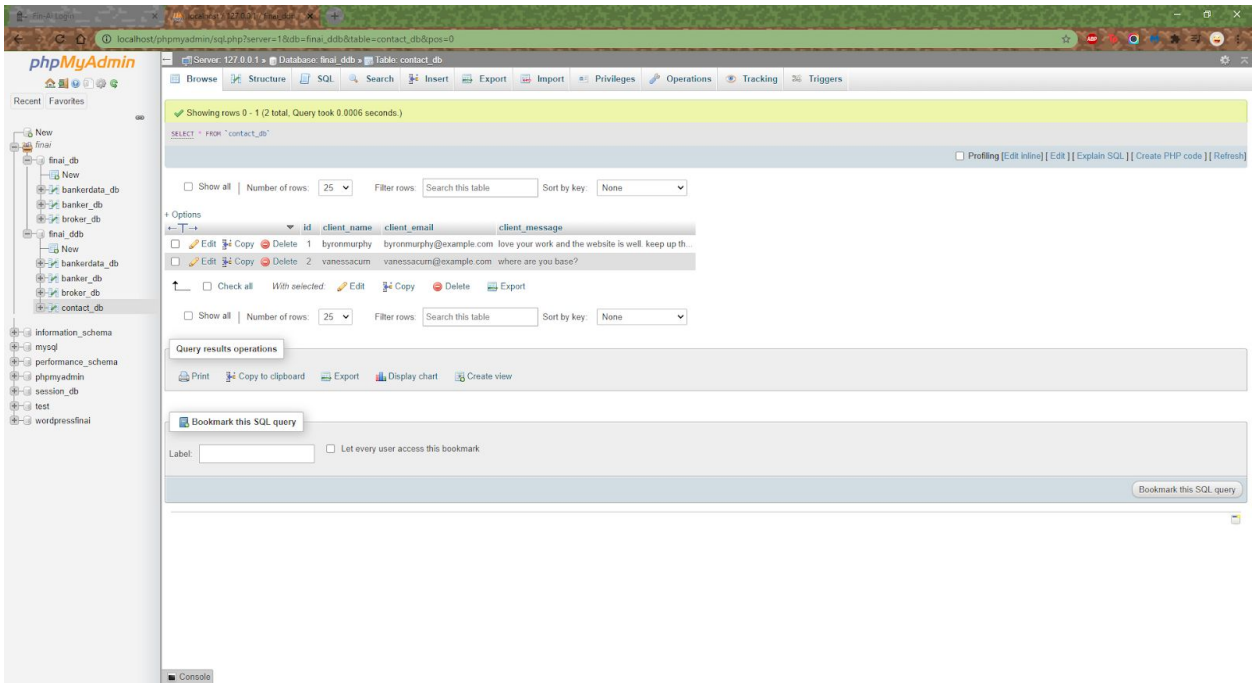
Broker Database Table



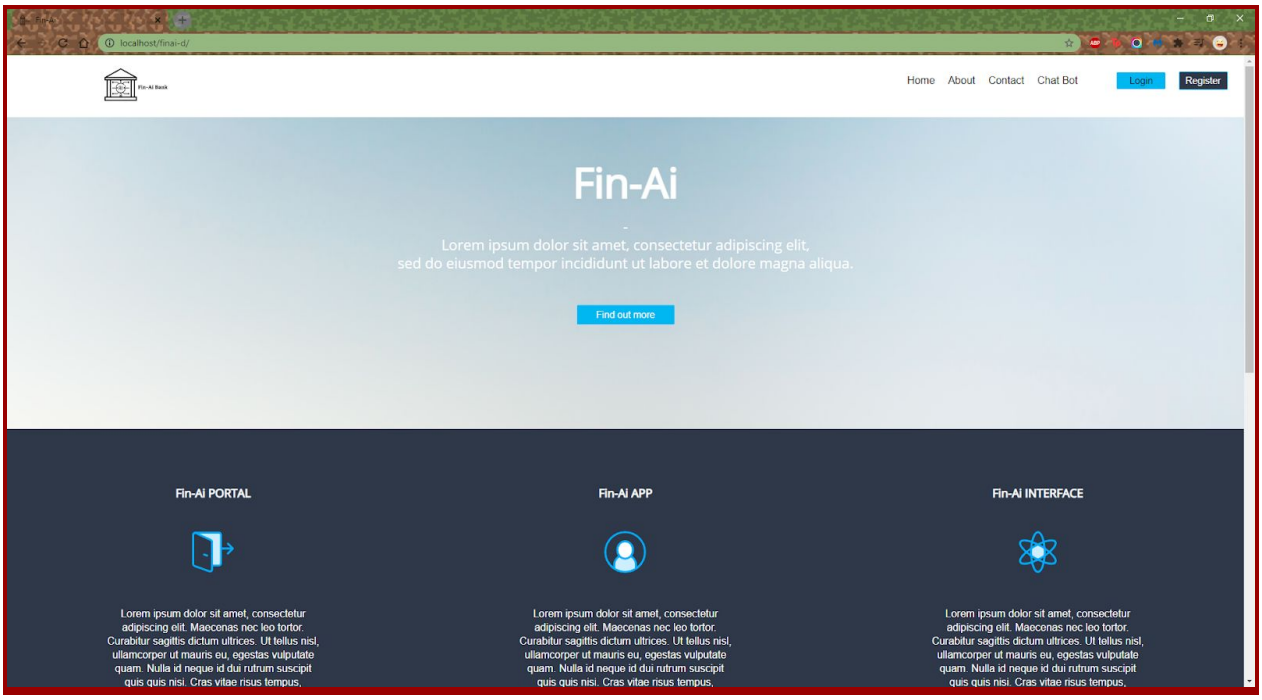
Banker Data Database Table



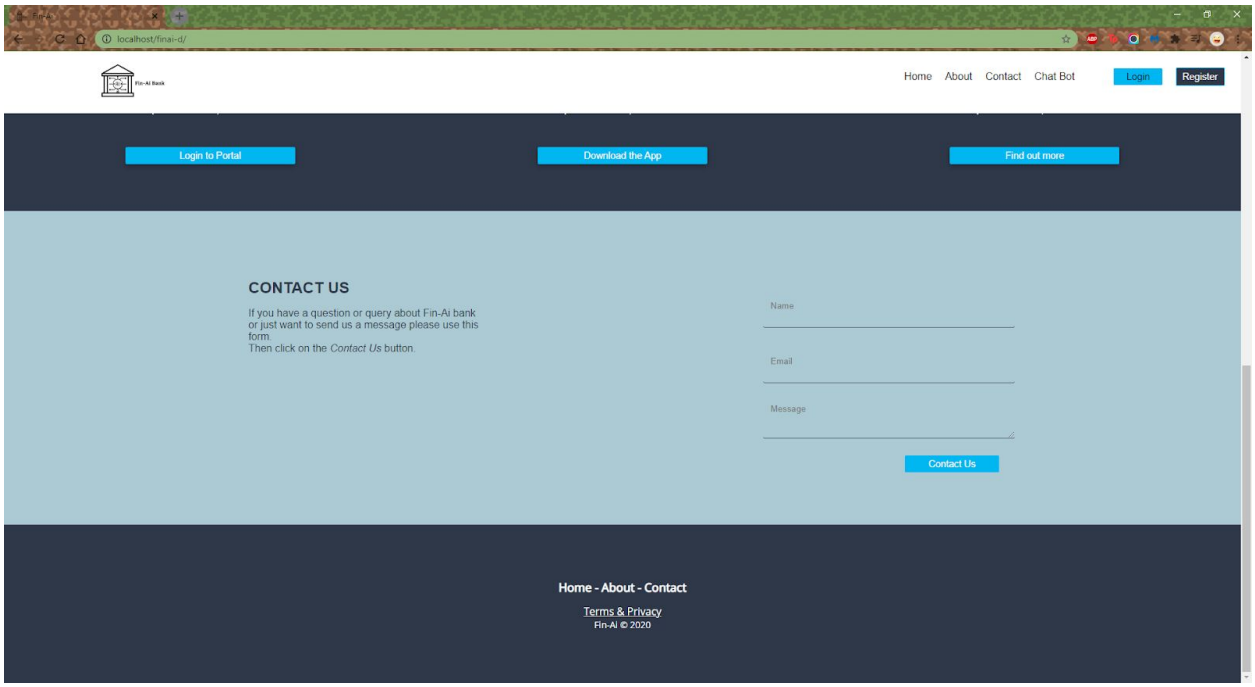
“Contact Us” database table



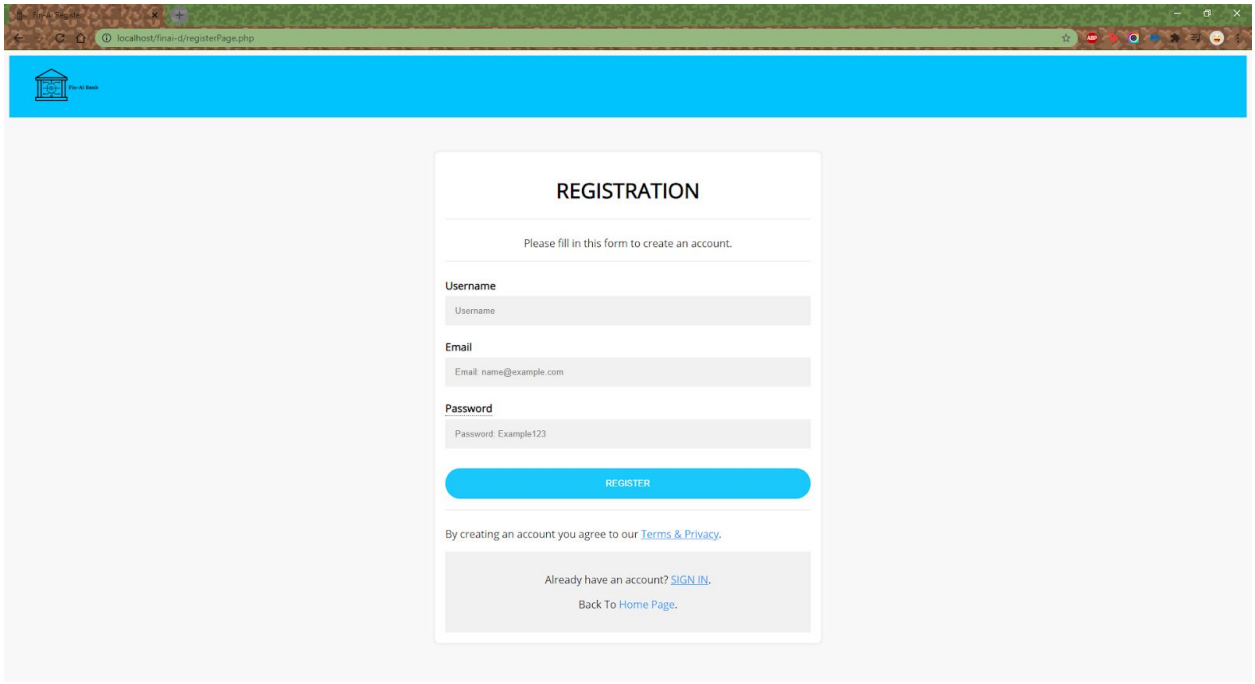
Website Home Page



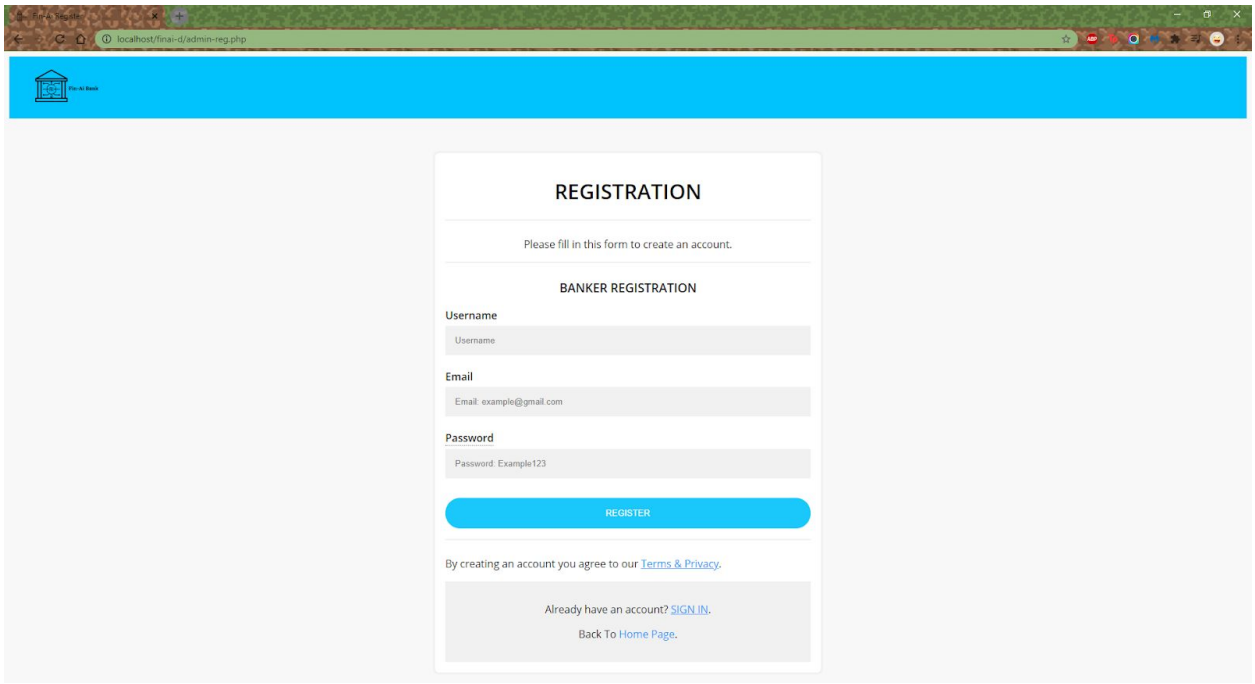
Website Contact Section (Home Page)



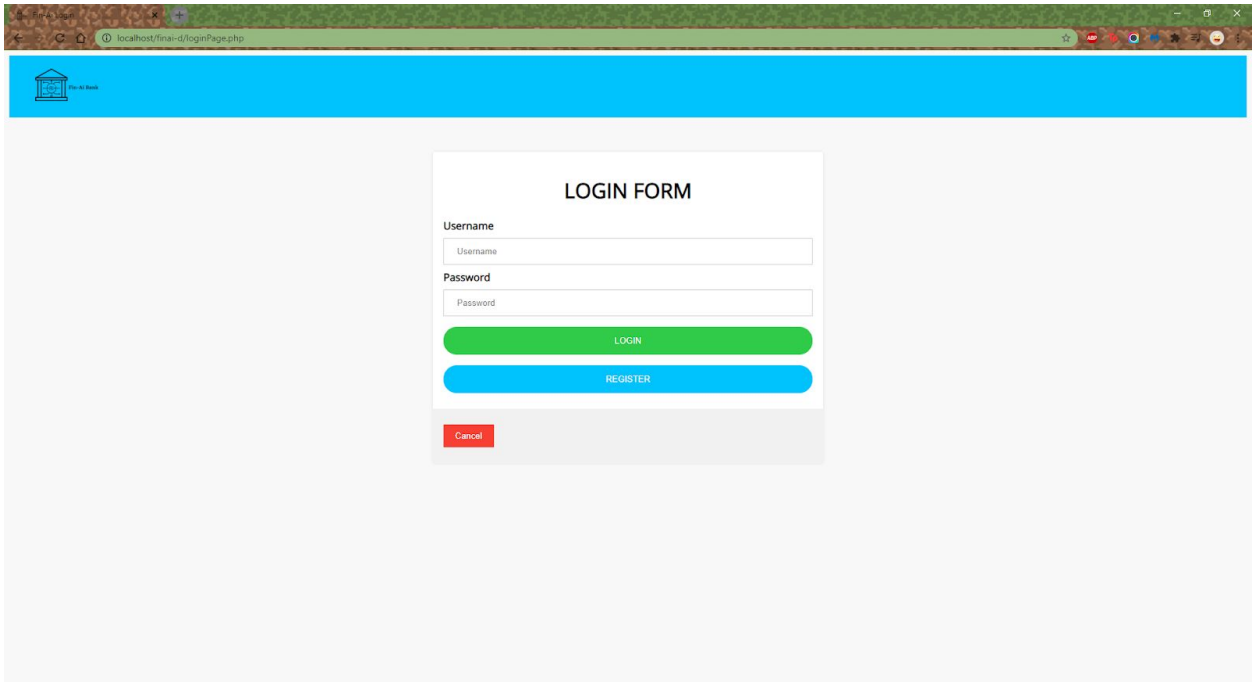
Website Registration - Client (Broker)



Website Registration - Banker



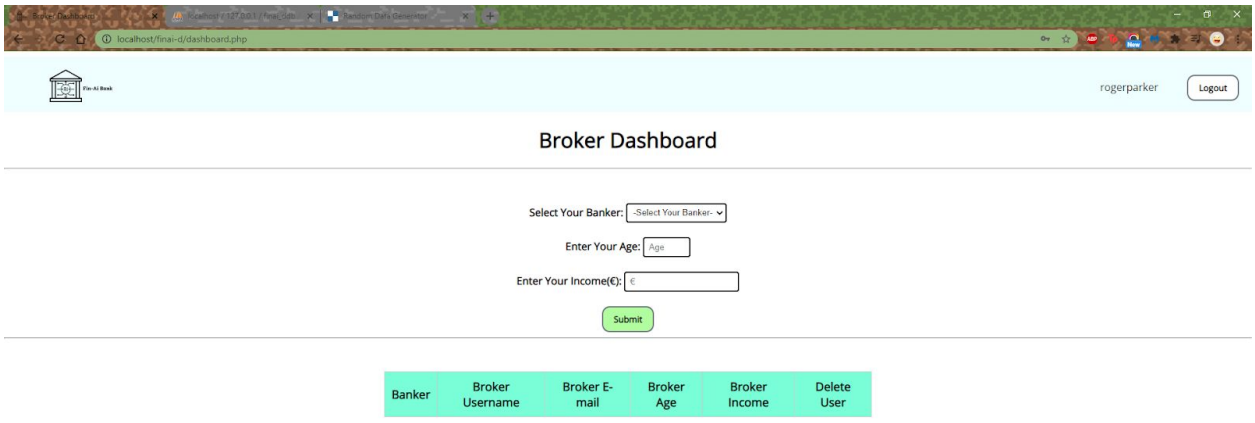
Website Login



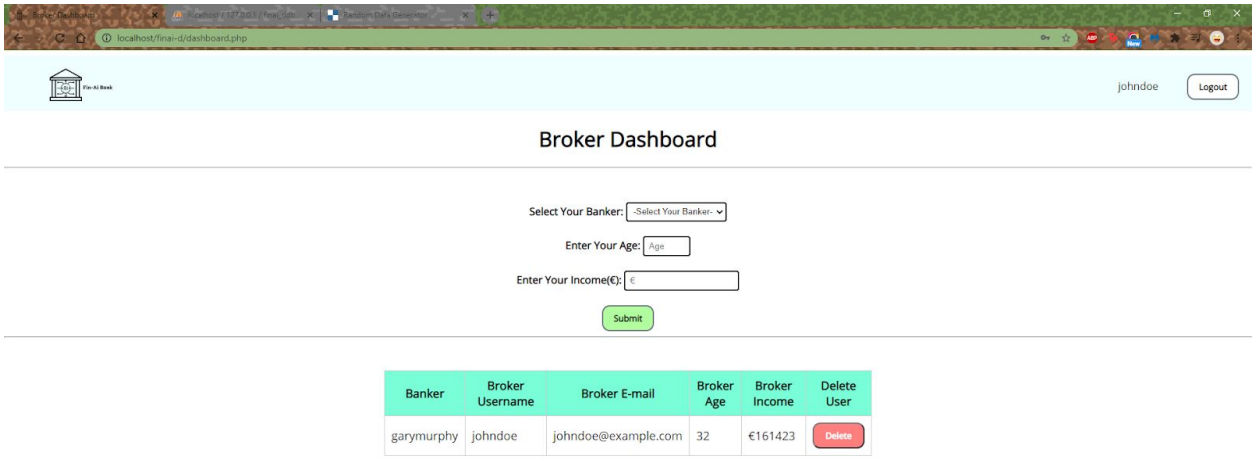
Website User Profiling

Broker Dashboard

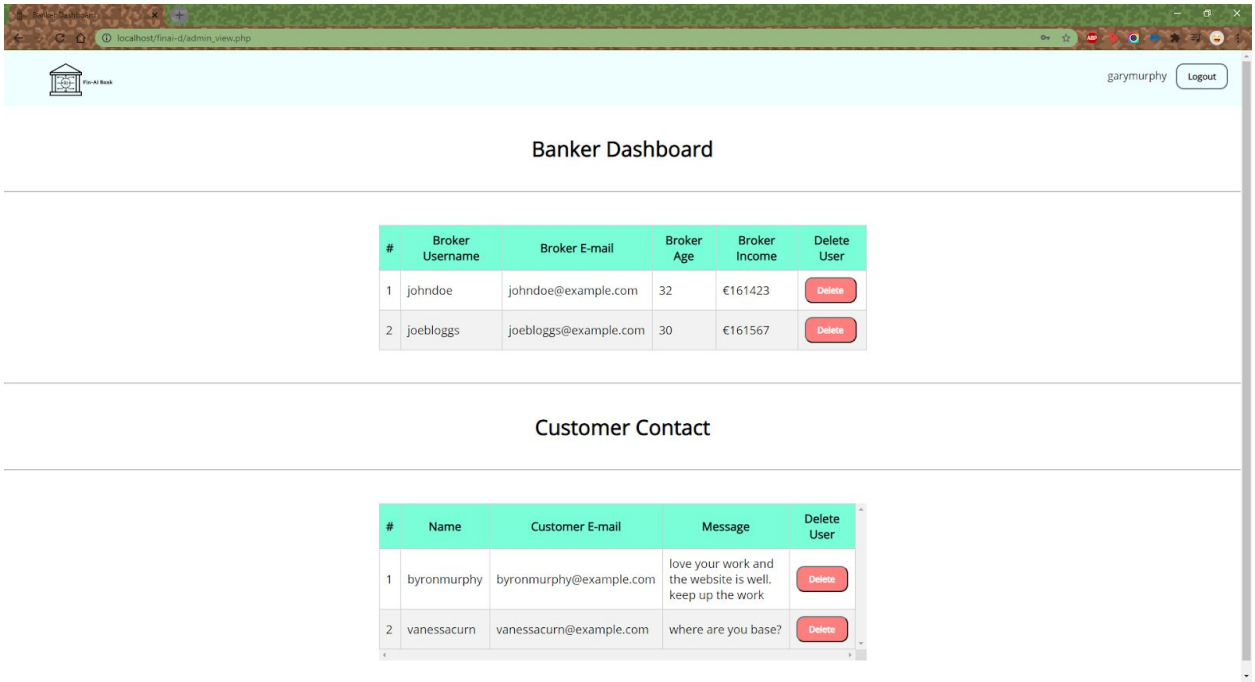
No User Data



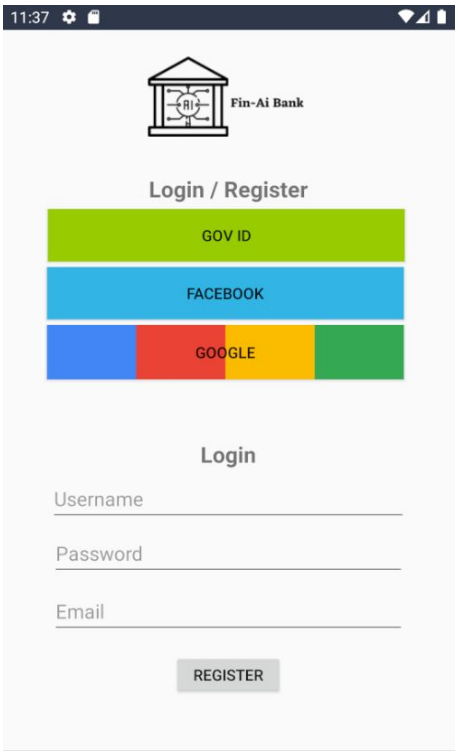
User Data



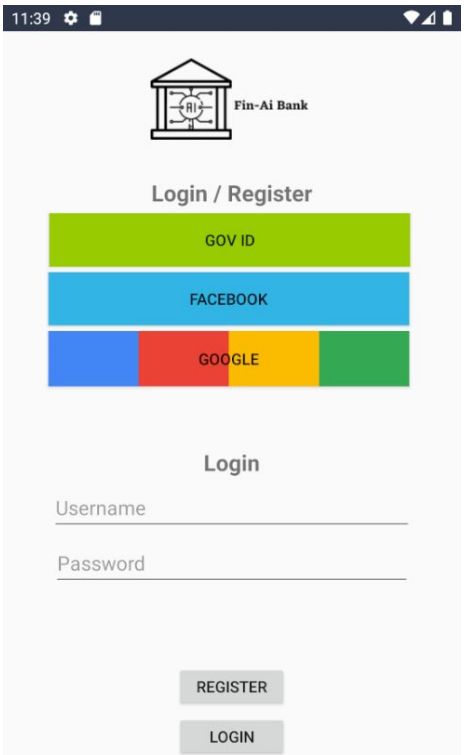
Banker Dashboard



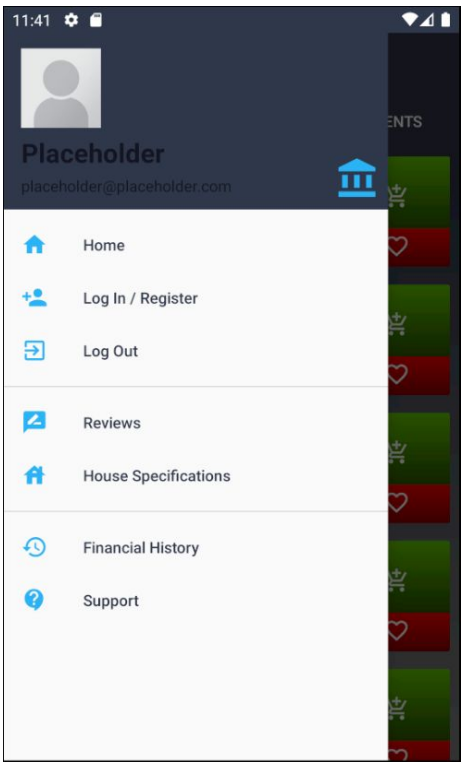
Mobile Registration



Mobile login



Mobile Navigation Menu



Sprint 3 (Nov 22 - Dec 23)

Sprint Backlog

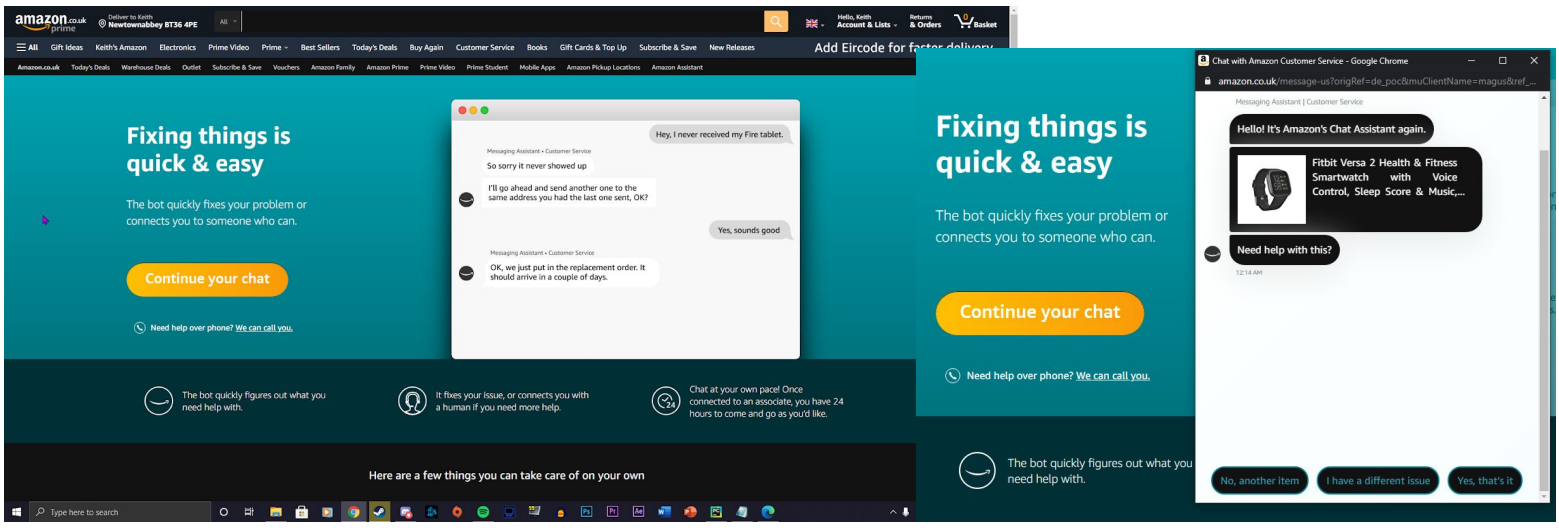
- Separate bankers and brokers registration for web application.
- Initial development of dashboards for bankers and broker for web application
- Modify database
- Mobile Tabbed Layout
- Mobile RecyclerView in tabbed fragments
- Mobile Connection to Database to allow signing in
- Begin development on Fin-AI Engine

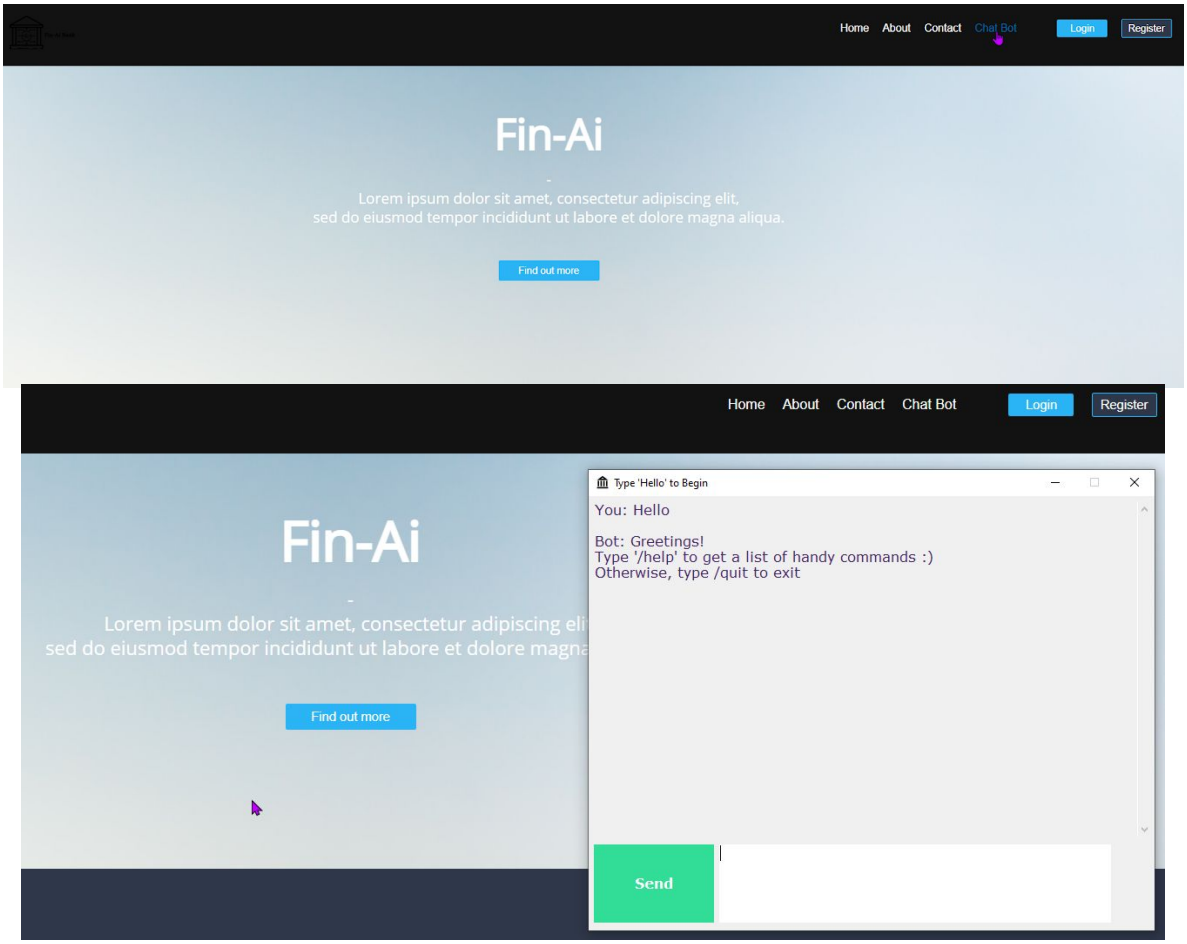
Sprint Details

Fin-AI Development

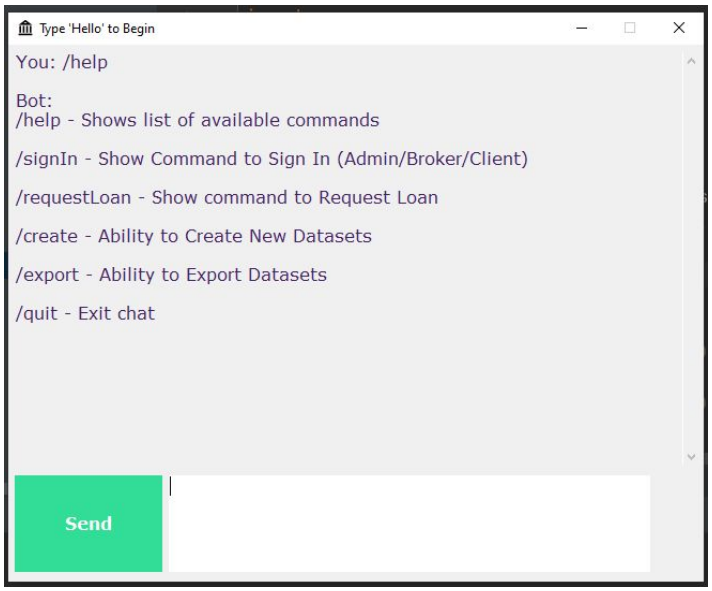
When issued the brief for this project, we learned each group required a custom ‘feature’, we decided to build this into the Fin-AI chatbot, where the chatbot would work something like some currently existing Customer Support applications online.

For example, with Amazon, when you click on ‘Customer Support’ and ‘Start Chat’, it boots you into a new window where the chat GUI is rendered. This is what we based our custom feature off of, where it is easy to navigate from the website to the chatbot with one click.





From within this chatbot, we have allowed users to sign in, and request loans, to see if they're eligible for loans or to purchase houses, these commands are easily accessible, by typing in '/help', and has friendly user responses should they enter invalid data, as opposed to crashing or not responding entirely.



Test Phase

Website

Tested (Working)

- Website integration with the database.
- Login and registration form.
- Appending data from the website to the database.
- Query data from the database and display data to the website.
- Session for each user.
- Different dashboard for each user.
- Form validation.
- Users' passwords are encrypted.
- Homepage contact section, public customer can message and query the bank.

Not Working

- Website Aggregation of New Client Data.
- "Delete" button to remove data from the database.
- Mobile and Fin-Ai engine chatbot integration.

App

Tested (Working)

- Database Connection using PHP scripts
- Login / Registration backend and frontend
- Decryption/Encryption of passwords in the database
- Basic Layout of the user interface

Not Working

- Content for the Prediction, Details and Payments tab
- Login with various types of social media accounts

Fin-AI Engine

Tested (Working)

- User Sign-In is working, data is pulled from database CSV file as opposed to connecting to database, as passwords have been encrypted on database, and Python was unable to decrypt them, so we decided to pull data from decrypted database CSV files instead.
- The AI pulls data from both provided datasets to predict the rough cost of a house, and rough amount of income overall, to predict the ability to get loans.
- Chatbot provides automated customer service, and is trained to respond to unknown messages/commands to an extent, instead of locking up or crashing.
- Chatbot GUI renders provided the local Python (Flask) server is running, when prompted by clicking ‘Chatbot’ on index.html.
- Reporting which displays the accuracy of loan/house suitability is printed within Python server terminal, as well as the average income, and average house price.

Not Working

- Unable to connect API to Phone App entirely, only connected to Web Application by means of Chatbot, cannot communicate otherwise.
- When exiting chatbot (by clicking ‘X’ or typing ‘/quit’), fails to render the fully customised ‘chatbot.html’ page, and only renders the bare .html page without CSS or JavaScript, thinking this is because the GUI and HTML template are hosted on a local server, and Python doesn’t allow us to host extra files, so we cannot additionally host the CSS to make the page look as intended.
- Unable to connect Python app to database, and passwords on database were encrypted, so instead, signing in is used by comparing the decrypted data on the database's CSV file.