



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

Fin-AI Cover Page	1
Table of Contents	2
Introduction	4
Project Overview	4
Goals of the Project	4
Group Roles	4
Database Overview	5
Tables	5
Table Details	5
Website Overview	7
Current Features	7
Technologies Used	7
Difficulties Encountered	8
Additional Features from Future Development	8
Android Application Overview	9
Current Features	9
Technologies Used	9
Difficulties Encountered	9
Additional Features from Future Development	9
Fin-AI Engine & Chatbot Overview	11
Current Features	11
Technologies Used	11
Difficulties Encountered	11
Additional Features from Future Development	11
Pre-Sprint Preparation	12
Sprints	13
Sprint 1 (Oct 16 - Nov 8)	13
Sprint Backlog	13
Sprint Details	13
Sprint 2 (Nov 8 - Nov 22)	15
Sprint Backlog	15

Sprint Details	15
Sprint 3 (Nov 22 - Dec 23)	23
Sprint Backlog	23
Sprint Details	24
Test Phase	26
Website	26
App	26
Fin-AI Engine	27

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

Phase 1	Subite...	People	Status	Timeline	Priority	
Create a remote repository to work on			Done	✓ Oct 8 - 9	Medium	
Decide + setup communication platform			Done	✓ Oct 5 - 7	Medium	
Document Product Back Log			Done	✓ Oct 5 - 9	Medium	
Document Product Overview			Done	✓ Oct 5 - 9	Medium	
MockUp & Prototype of Web & Mobile App		KB	Done	✓ Oct 5 - 16	Medium	
Submit Week 03 Deliverables			Done	✓ Oct 17 - 18	Critical	

Phase 2 Prelude	Subite...	People	Status	Timeline	Priority	
Research and Documentation of Web, Mobile App & machine learning AI		+3	Done	✓ Oct 16 - 21	Medium	
Create a Sprint or Phase Map. Task Map			Done	✓ Oct 17 - 20	Medium	
Implement Database model			Done	✓ Oct 18 - 24	Medium	
Negotiate the IDE or platform to be use for the project.		+3	Done	✓ Oct 16 - 21	Medium	

Sprints

Sprint 1 (Oct 16 - Nov 8)

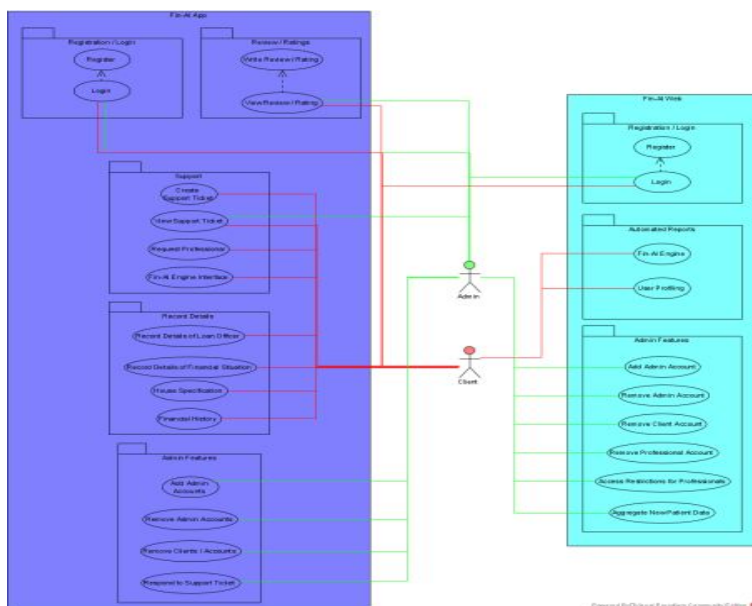
Sprint Backlog

- 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

Phase 2 - Sprint 1	Subt...	People	Status	Timeline	Priority
Decide production environment of website + create basic template	🗨	👤	Done	Nov 8	High
Website Use Cases	🗨	👤	Done	Nov 8	Medium
Website User Stories	🗨	👤	Done	Nov 8	Medium
Sprint retrospective/review	🗨	👤	Done	Nov 8	Medium
Submit Sprint 1 Deliverables	🗨	👤	Done	Nov 8	Critical
+ Add					

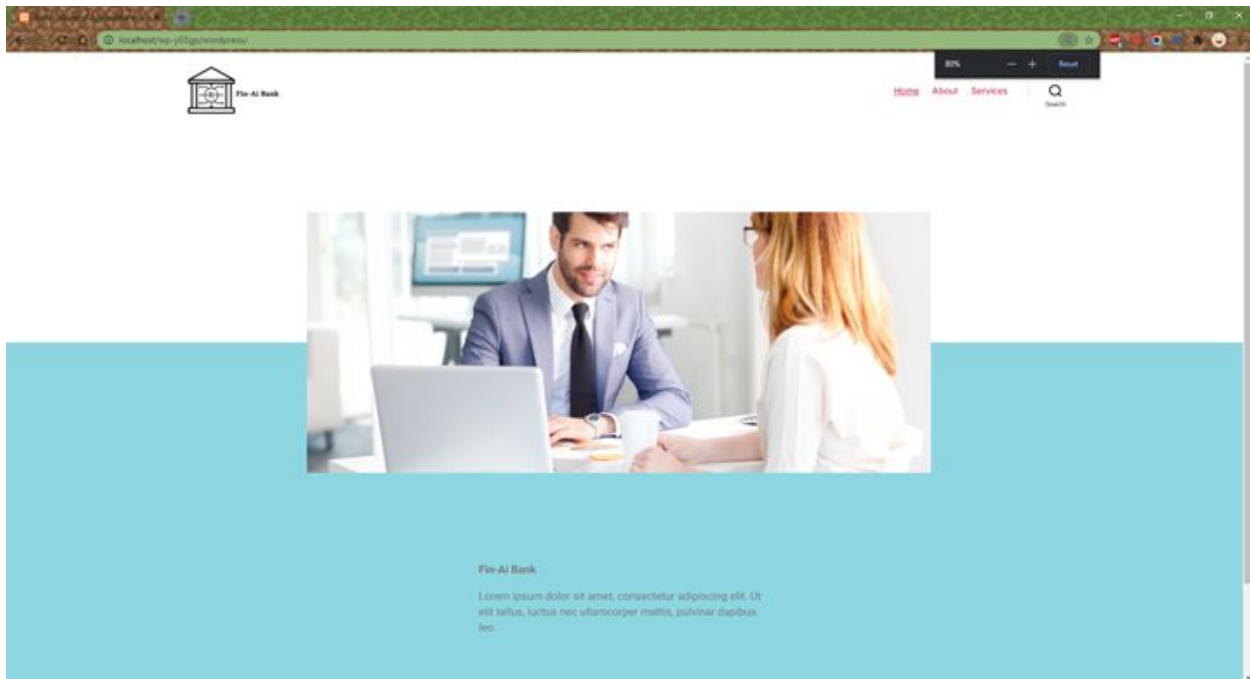
The use of a 'Monday.com' board proved to be a much more efficient project management tool for our group than that of a Trello board. It is easy to assign certain tasks and their due dates to each team member. Unlike with a Trello board, all important information is already displayed in a much more simpler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

Phase 2 - Sprint 2	Subite...	People	Status	Timeline	Priority	
Continue Development of Web	3		Done	Nov 8 - 22	Medium	
Begin development of app	5		Done	Nov 8 - 22	High	
App: Use Cases		KB	Done	Nov 12	Medium	
App: Use Case User Stories			Done	Nov 12	Medium	
Sprint retrospective/review			Done	Nov 22	Medium	
Submit sprint 2 deliverables			Done	Nov 22	Critical	
+ Add						

Database Banker Database table

The screenshot shows the phpMyAdmin interface for the 'final_db' database, specifically the 'banker_db' table. The table structure is defined as follows:

id	admin_username	admin_email	admin_pwd
1	garymurphy	garymurphy@example.com	\$2y\$10\$2H2658KQH05.mp2x.5u1uUFCc03wJvQFE8a1qGTXd...
2	marybarry	marybarry@example.com	\$2y\$10\$zcp1TjBhHqjLViGoeb84cCr1dAOVHoC60Qdc...

The interface also shows the 'Query results operations' section with options to print, copy to clipboard, export, display chart, and create view. A 'Bookmark this SQL query' section is also visible.

Broker Database Table

The screenshot shows the phpMyAdmin interface for the 'final_db' database, specifically the 'broker_db' table. The table structure is defined as follows:

id	username	user_email	user_pwd
1	janedoe	janedoe@example.com	\$2y\$10\$VQDmMzzlumOqApA.Gsbl.uanSXTlpsjTndByNL.oxzZH...
2	joebloggs	joebloggs@example.com	\$2y\$10\$7Of8cOTE7day1sAIBaImqQZKoaasD.ngcOP0H4BHO9tq...
3	jenameza	jenameza@example.com	\$2y\$10\$CpPVjU0hS0ADmewkew9dQppYEEYanCyAIsAMovedFv...
4	zaidalman	zaidalman@example.com	\$2y\$10\$E..tmK9KdvzcQjccWqj.cGxTRUabAulCk8dCfRtx...

The interface also shows the 'Query results operations' section with options to print, copy to clipboard, export, display chart, and create view. A 'Bookmark this SQL query' section is also visible.

Banker Data Database Table

The screenshot shows the phpMyAdmin interface for the 'final_db' database. The 'bankerdata_db' table is selected, displaying 4 rows. The table structure is as follows:

	id	admin_username	broker_username	broker_email	broker_age	broker_income
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	garymurphy	janedoe	janedoe@example.com	30	161423
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	garymurphy	zaidalman	zaidalman@example.com	34	171113
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	marybarry	jenameza	jenameza@example.com	29	157800
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	marybarry	joebloggs	joebloggs@example.com	36	161500

Below the table, there are options for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and a 'Bookmark this SQL query' section with a label input field and a checkbox 'Let every user access this bookmark'.

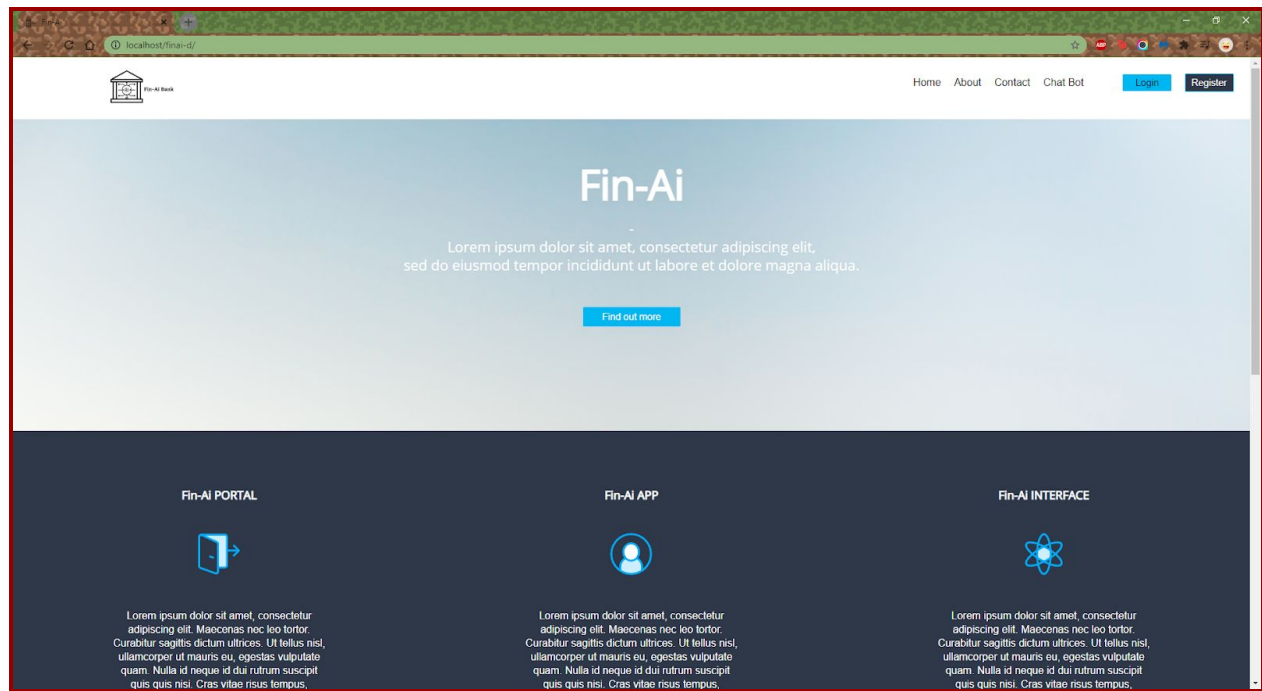
“Contact Us” database table

The screenshot shows the phpMyAdmin interface for the 'final_db' database. The 'contact_db' table is selected, displaying 2 rows. The table structure is as follows:

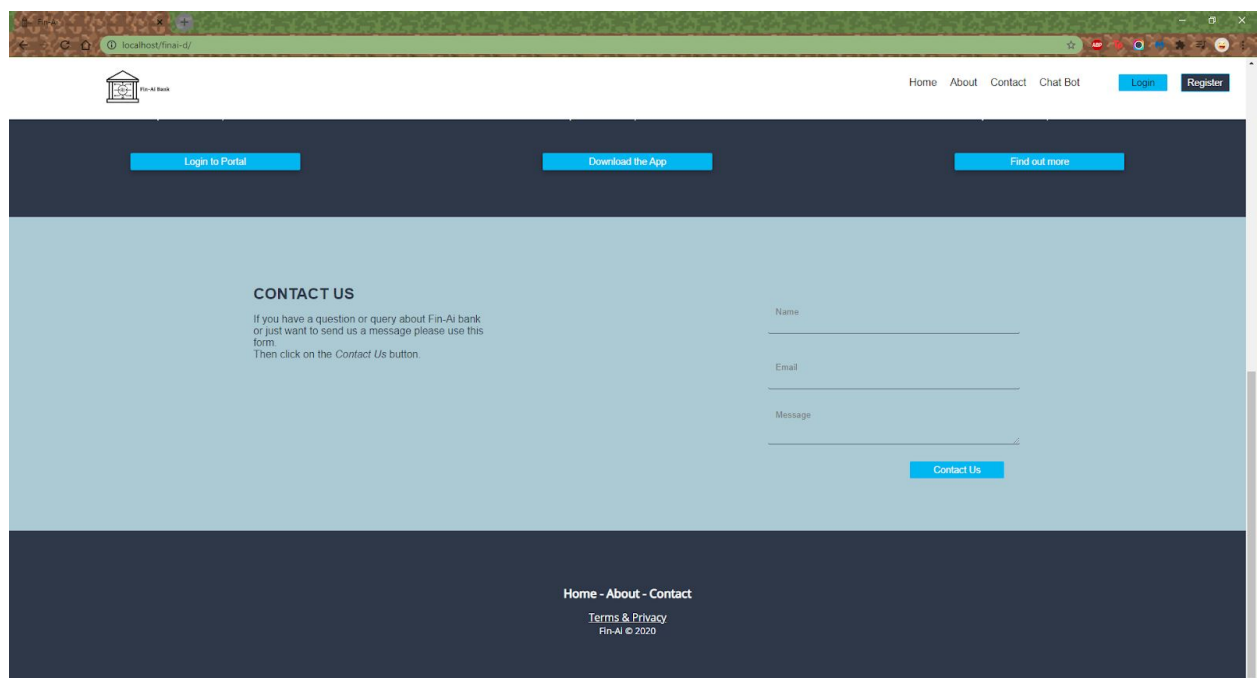
	id	cclient_name	cclient_email	cclient_message
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	byronmurphy	byronmurphy@example.com	love your work and the website is well keep up th
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	vanessacum	vanessacum@example.com	where are you base?

Below the table, there are options for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and a 'Bookmark this SQL query' section with a label input field and a checkbox 'Let every user access this bookmark'.

Website Home Page



Website Contact Section (Home Page)



Website Registration - Client (Broker)

REGISTRATION

Please fill in this form to create an account.

Username
Username

Email
Email: name@example.com

Password
Password: Example123

[REGISTER](#)

By creating an account you agree to our [Terms & Privacy](#).

Already have an account? [SIGN IN](#).

[Back To Home Page](#).

Website Registration - Banker

REGISTRATION

Please fill in this form to create an account.

BANKER REGISTRATION

Username
Username

Email
Email: example@gmail.com

Password
Password: Example123

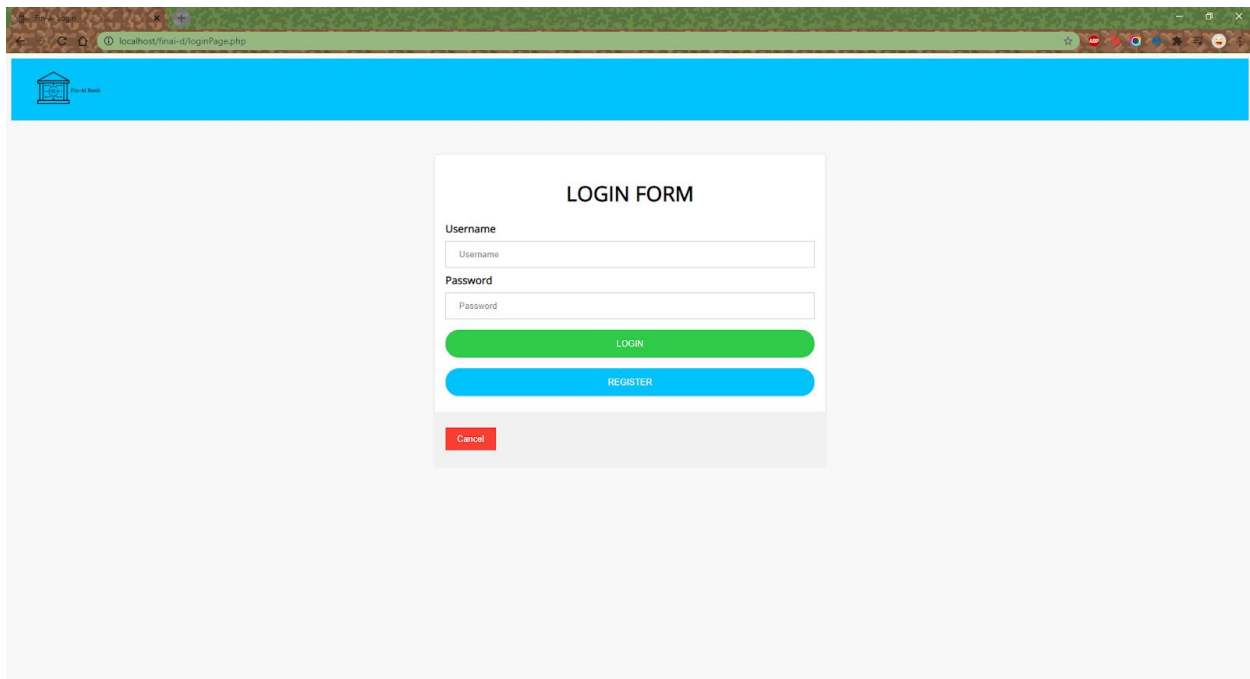
[REGISTER](#)

By creating an account you agree to our [Terms & Privacy](#).

Already have an account? [SIGN IN](#).

[Back To Home Page](#).

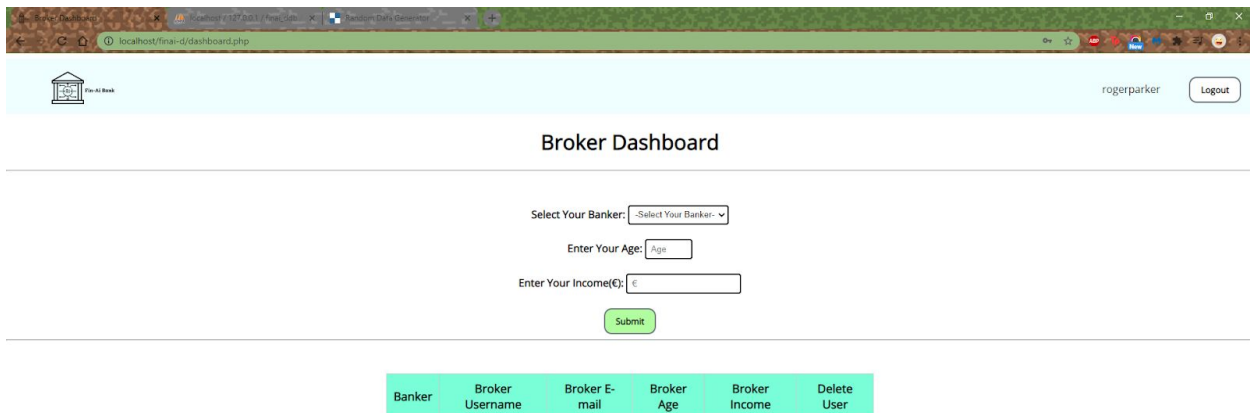
Website Login



A screenshot of a web browser showing a login page. The browser's address bar displays 'localhost/finan-d/loginPage.php'. The page has a blue header with a logo and the text 'Finan-D Bank'. The main content area is light gray and contains a white box titled 'LOGIN FORM'. Inside this box, there are two input fields: 'Username' and 'Password'. Below these fields are three buttons: a green 'LOGIN' button, a blue 'REGISTER' button, and a red 'Cancel' button.

Website User Profiling

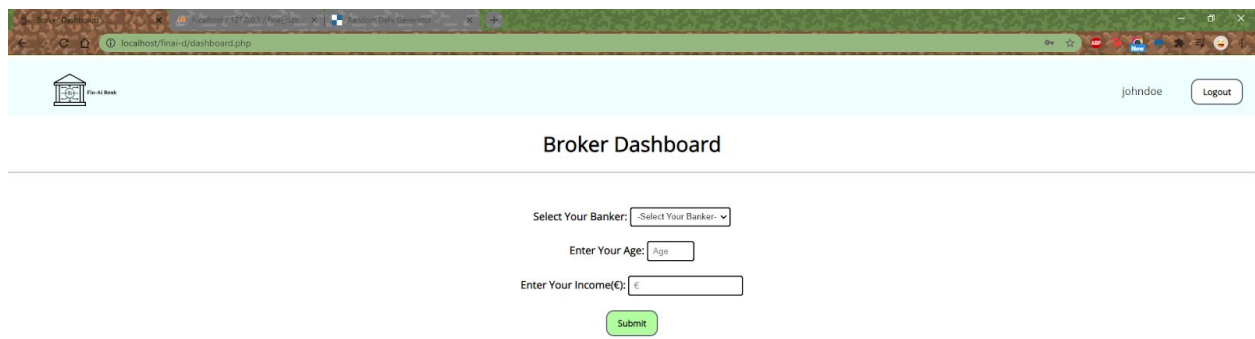
Broker Dashboard No User Data



A screenshot of a web browser showing a broker dashboard. The browser's address bar displays 'localhost/finan-d/dashboard.php'. The page has a light blue header with a logo, the text 'Finan-D Bank', and a user profile section showing 'rogerparker' and a 'Logout' button. The main content area is white and contains a form titled 'Broker Dashboard'. The form has three input fields: 'Select Your Banker:' with a dropdown menu, 'Enter Your Age:' with a text input, and 'Enter Your Income(€):' with a text input. Below these fields is a green 'Submit' button. At the bottom of the page, there is a table with five columns: 'Banker', 'Broker Username', 'Broker E-mail', 'Broker Age', 'Broker Income', and 'Delete User'.

Banker	Broker Username	Broker E-mail	Broker Age	Broker Income	Delete User
--------	-----------------	---------------	------------	---------------	-------------

User Data



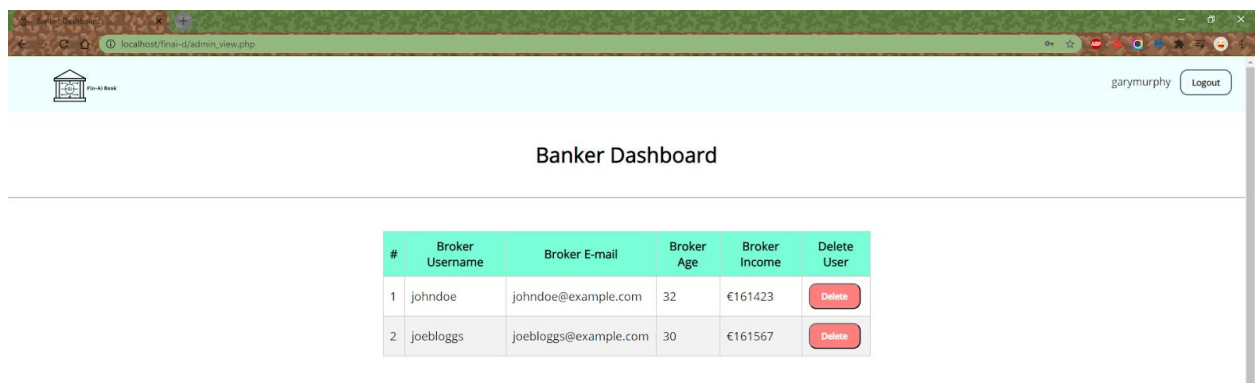
Select Your Banker:

Enter Your Age:

Enter Your Income(€):

Banker	Broker Username	Broker E-mail	Broker Age	Broker Income	Delete User
garymurphy	johndoe	johndoe@example.com	32	€161423	<input type="button" value="Delete"/>

Banker Dashboard

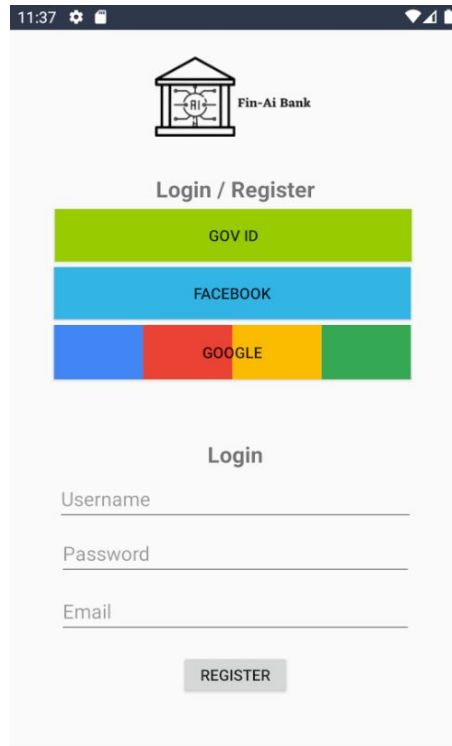


#	Broker Username	Broker E-mail	Broker Age	Broker Income	Delete User
1	johndoe	johndoe@example.com	32	€161423	<input type="button" value="Delete"/>
2	joebloggs	joebloggs@example.com	30	€161567	<input type="button" value="Delete"/>

Customer Contact

#	Name	Customer E-mail	Message	Delete User
1	byronmurphy	byronmurphy@example.com	love your work and the website is well. keep up the work	<input type="button" value="Delete"/>
2	vanessacurn	vanessacurn@example.com	where are you base?	<input type="button" value="Delete"/>

Mobile Registration



11:37

Fin-Ai Bank

Login / Register

GOV ID

FACEBOOK

GOOGLE

Login

Username

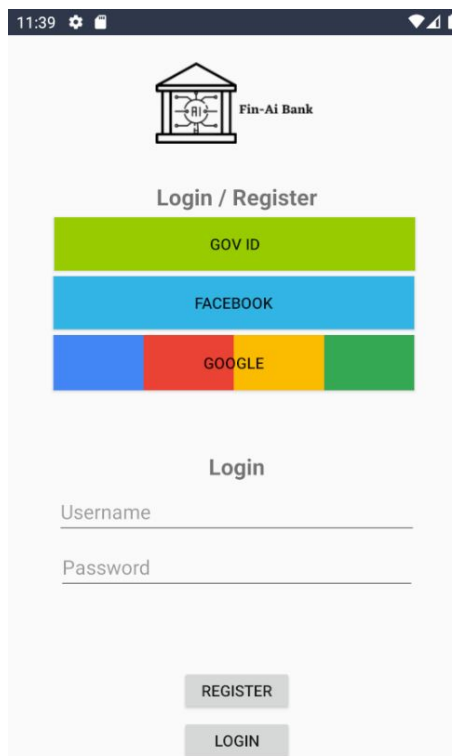
Password

Email

REGISTER

This screenshot shows the mobile registration interface for Fin-Ai Bank. At the top, the status bar displays the time 11:37. The app header features the Fin-Ai Bank logo and name. Below the header, the 'Login / Register' section contains three large, colorful buttons for 'GOV ID' (green), 'FACEBOOK' (blue), and 'GOOGLE' (red, yellow, and green). The 'Login' section below has three input fields for 'Username', 'Password', and 'Email', followed by a 'REGISTER' button.

Mobile login



11:39

Fin-Ai Bank

Login / Register

GOV ID

FACEBOOK

GOOGLE

Login

Username

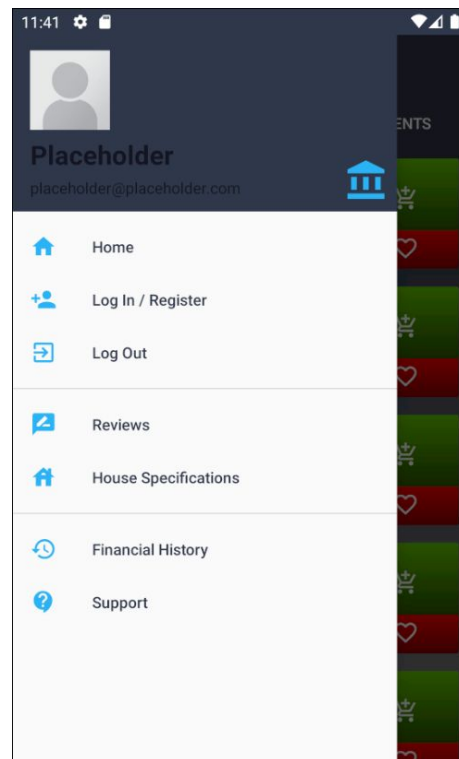
Password

REGISTER

LOGIN

This screenshot shows the mobile login interface for Fin-Ai Bank. At the top, the status bar displays the time 11:39. The app header features the Fin-Ai Bank logo and name. Below the header, the 'Login / Register' section contains three large, colorful buttons for 'GOV ID' (green), 'FACEBOOK' (blue), and 'GOOGLE' (red, yellow, and green). The 'Login' section below has two input fields for 'Username' and 'Password', followed by 'REGISTER' and 'LOGIN' buttons.

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

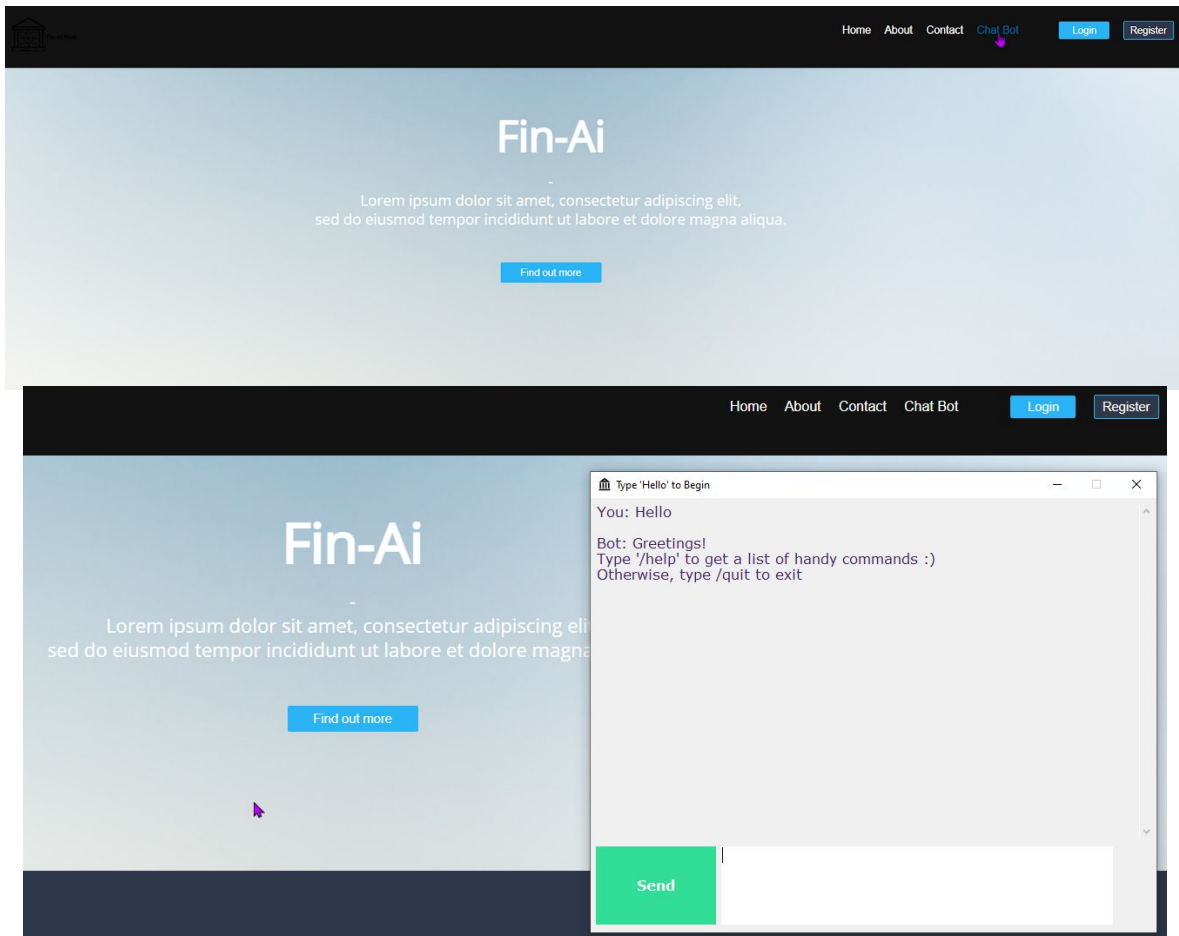
Phase 2 - Sprint 3	Subite...	People	Status	Timeline	Priority	
Continue mobile app	5		Done	✓ Nov 25 - Dec 13	Medium	
Continue web app	2		Done	✓ Nov 25 - Dec 13	Medium	
Begin Fin-AI Engine	1	KB	Done	✓ Nov 25 - Dec 13	High	
Sprint retrospective/review	1		Done	✓ Nov 25 - Dec 13	High	
Submit sprint 3 deliverables	3		Done	✓ Nov 25 - Dec 13	Critical	

Fin-AI Development

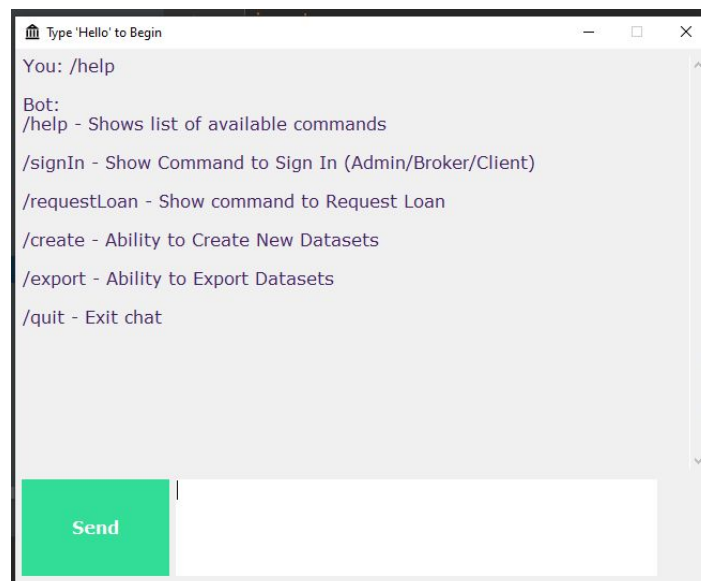
When issued the brief for this project, we learned each group required a custom ‘feature’, we decided to build this into the Fin-AI chatbot, where the chatbot would work something like some currently existing Customer Support applications online.

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.

The collage consists of three screenshots from the Amazon website and chat interface. The top left screenshot shows the Amazon.co.uk homepage with a teal banner that says "Fixing things is quick & easy" and a "Continue your chat" button. The top right screenshot shows a chat window titled "Chat with Amazon Customer Service - Google Chrome" with a "Hello! It's Amazon's Chat Assistant again." message and a "Continue your chat" button. The bottom screenshot shows a chat window titled "Chat with Amazon Customer Service - Google Chrome" with a "Hello! It's Amazon's Chat Assistant again." message and a "Continue your chat" button.



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.