PYTHON APPLICATION

Peer-to-Peer Social Media Network

Amal Chandra
[COMPANY NAME] [Company address]

Abstract

The aim of this project was to create a peer-to-peer social media network for a company's executive team to communicate with each other without fear of being monitored. The developed application must be secure due to privacy concerns and to prevent unwanted monitoring of their communications by rival companies. The application was made using Python for the backend, HTML/CSS for the frontend and all the data were stored in the SQLite3 database. The purpose of this project was to learn Networking, Web Development, Security, and Software Project Planning & Development.

Requirements:

The messaging service application has been developed for the client which uses login server network model for authentication purpose and peer-to-peer network for sending messages and file this is a hybrid network. The most important requirement of this project is that the application must be secure. Therefore, requirements from the client was to develop a system where the users must log into the system to be able to use the application. The logged in users should be able to find all the other users online. All the users should be able to create and maintain a simple profile page, as well as request profile of other logged in users. The users should also be able to easily send messages and files (images, audio and PDF) to other online users.

The above requirements were met as the developed system allows the user to login to the system by entering their username and password from anywhere in the world. Two-Factor Authentication has also been implemented therefore the user must enter a secret code from their phone which changes every 30 seconds. The system has been developed to make it easy for the users to find all the users as they are all sorted from online users to offline users, so that they can easily start a conversation with the online users. The users can send messages and files (images, audio and PDF) which are embedded therefore making it easier for them to see the content of the file. The users can create and view a profile page which shows their picture, full name, position, description, location and last updated. All this user information is stored in the database.

Top-Level View of The System

The diagram in the appendix (1) gives a top-level view of how the system works. The login server is used to authenticate the users detail. This returns a message to the user letting them know they have successfully logged in. Once the users are logged in they can send messages and files to other online users and view another online users' profile. This is a peer-to-peer network; therefore, the messages and files are sent straight to the user without going to a central or login server. The users interact with the HTML/CSS which using CherryPy passes the information to the python where the logic occurs. The python gives feedback back to the users which is displayed in HTML. Python interacts with the database to store and get information for the users.

Development Issues

There were many issues that I faced during the development of the messaging service application, but the two significant issues that I faced were that I was not able to easily communicate between the frontend (web page) and the backend (python) software. This was a significant issue for me as I never had a project where I had to deal with both the frontend and the backend. After doing some research I found out I could use JavaScript to overcome this issue. Therefore, I spent some time researching and asking questions to the TA regarding JavaScript. After spending several hours trying to implement it I was unsuccessful. Since we had very limited time to do the project I decided to find other ways to interact with the frontend and the backend. Therefore, I used Jinja2 to do the templating of my website. This allowed me to easily pass variables from python to the HTML to be displayed to the users. Jinja2 also allowed me to do basic logic in my templates rather than doing it in Python. To pass inputs from the user (HTML) to the python I used a simple HTML form to do this. Therefore, the user enters their detail and with a press of the button submits the information. Due to not being able to use JavaScript I was unable to do auto-refreshing of the website which would have been used to refresh online users.

The other significant issue that I faced during the development of the messaging service application was unable to understand how the database works. This was because database was a new concept for me. This caused me issues as I wasn't aware how python and database work together. Therefore, I had to learn what a database is and how it works with Python. Since database and Python were both new to me I didn't know any of the syntax to create, insert and update database table. To overcome this issue, I watched online tutorials and asked the TA's for help. While I was doing research on how to use database I came across SQL injection. I searched online on how to prevent it, from which I found the best way to avoid SQL injection is to escape the user input. This did not make any sense to me therefore I asked the TA what it means and how to escape the user inputs. The other issue that I incurred due to being a beginner in database is that my database would get locked and I had no idea on how to fix this, therefore, at the beginning I would close all my program to fix this issue, but I later found out from the TA that this is cause by my database being accessed from the DB Browser and the Python code and closing the Database would fix the issue. Using the database, researching and asking the TA's for help with database helped me learn a lot about how to use database with Python which saved me a lot of time as the project progressed.

Features that Improve Functionality of the System

The application has been developed with additional features rather than just meeting the minimum requirements. Since the main purpose of this project was to make an application for a company who had concerns with security, the application has been made with additional security features. This includes Two Factor Authentication, which means the user needs to enter their normal login details (username and password) as well as a secret code from their Google Authenticator app which changes every 30 seconds. Therefore, the user always needs their phone with them to be able to access the messaging application. With Two Factor Authentication even if a hacker manages to get a user's password or if the password gets leaked, without the secret code they won't be able to sign in. The other security feature that has been implemented is protection from HTML and SQL injection, this means that no one can inject scripts to the application which could make the HTML misbehave or do SQL injection which could leak or corrupt crucial information from the database.

The minimum requirement required a functional prototype of the backend with the UI and UX as a secondary requirement. But the developed application had both the frontend and the backend as a primary requirement for the developers from the beginning. Therefore, the developed application has a nice and simple UI and UX. The users can simply find all the online users and start a conversation, as well as simply find any users' profile. The user needs are always a click away meaning the user do not have to type in the name of the other user in an input box, all the user has to do to start a conversation is to click the other users name and all their message history pops up with embedded viewers for images, videos and PDF. When the user sends a message or a file the application lets them know if the message/file was delivered successfully or not. The application has been texted in multiple different browsers such as Google Chrome, Firefox and Microsoft Edge.

The application has Unicode support which allows the users to express themselves in more than just normal English. The users can send emoji as well for the messages which could not simply be said in a normal language.

The application stores all the information in a database. Using a database makes it easier to manage all this information and makes it a bit secure compared to storing the information in a plain txt file.

Peer-to-Peer Suitability

Peer-to-peer network is created when multiple computers are connected and share resources without a need for a separate server. The developed messaging application uses a hybrid network which uses peer-to-peer network for the exchange of information between the users and a login server for authentication purpose. This is suitable for this project, as a central server is expensive to setup and maintain where as a peer-to-peer network is much cheaper and easier to setup. Since this project is for a small scale of less than 100 users peer-to-peer is more suitable. There are advantages and disadvantages of peer-to-peer network. The advantage as previously mentioned is that it is easier and cheaper to setup, it directly communicates with the intended node therefore doesn't have to rely on a central server. Having a central server would mean that if the central server goes down all communication would break therefore won't be able to send messages or files. The disadvantage of hybrid network is that it is not as secure as central server. As the information are stored locally in the computer. Therefore, if the information is not encrypted it can easily be accessed by another user, making it not secure. Therefore, the network we used for the application was suitable for this project.

Protocol Suitability

A network protocol describes the rules and conventions for communication between network devices. The application protocol for this project was designed by the students doing this project whereas the login server protocol was provided by the lecturer. The protocol was designed to be simple yet functional. The protocols were agreed upon by all the students, this needs to be agreed by all the students using it so that the messaging application works with all the different messaging application designed by other students. The protocol was suitable for this project. This is because the protocol provided suitable functionality to easily implement the requirements and also allowed to easily add optional features which could be used by the users who uses it and can be easily avoided if the features aren't used.

Suitability of Tools

The tools that was used to develop the messaging service application was Python for backend. Python is a suitable as it is a high-level programming language. It allows to write easy and readable code as the syntax are simpler compared to other programming language such as Java and C++. The other reason Python was suitable for this project is that information and help can be easily found in the internet, making it easier for people to develop programs. PyCharm was used as the Python IDE to develop the code. This is because PyCharm provides features such as code completion and type hinting making it easier to write python code. CherryPy was used for object-oriented web framework. This is because it was easy to setup and configure.

HTML/CSS was used for the frontend. This was used to design user interface for the users to interact with. HTML/CSS was suitable for this project as no additional downloads are required from the users and can be accessed from anywhere the internet is provided compared to having a separate application which needs to be downloaded by the user. HTML/CSS is also easy to learn, and as Python was the focus of this project less time was spent to learn and develop the frontend using HTML/CSS. HTML/CSS is widely used therefore easier to find information online.

Bitbucket was used for backup and version control. This ensures that the code is always safe in the cloud even if the file in the computer gets corrupted. Bitbucket also provides version control therefore allowing to easily go back to a working version of the code if something goes wrong with the code while developing.

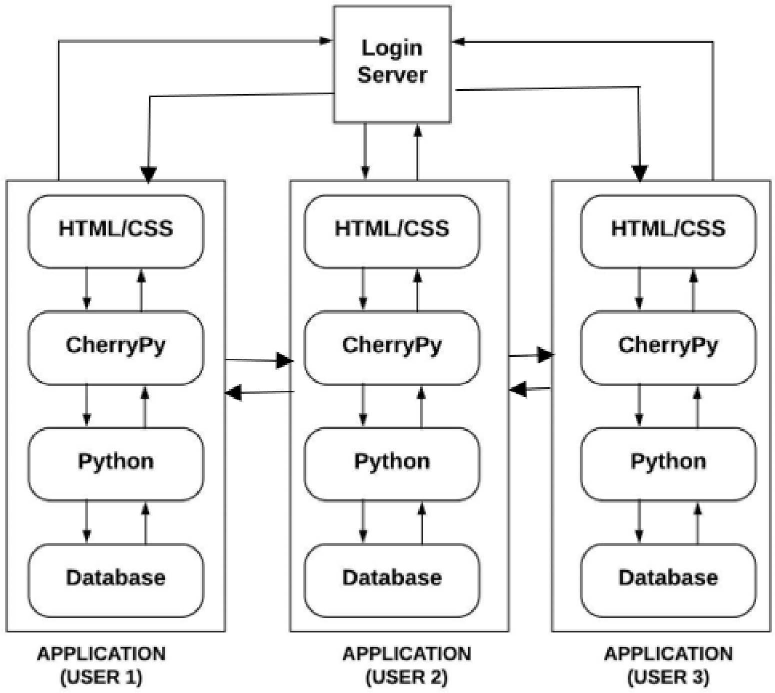
Suggested Improvements for Future Development

A lot of improvements can be made to the application. Despite the extension provided for the due date a lot of planned features for the application was not implemented due to the time constraints. The main focus of this application development was security. Therefore, in the future encryption will be added. The encryption would support standards 0 to 5, making the application very secure. The other security improvement that will be made would be to continue working on acknowledging the messages and files. Due to time constraints the acknowledge function had to be dropped as it was not fully developed. Acknowledge would ensure the receiver that the message they receive is the correct message and has not been tempered with. The other improvement that would be made to the application is in the front end. The user interface and user experience of the application would be improved. This would be done by making the website more dynamic, proving auto refresh of online users and messages, therefore the users won't have to always refresh the page to find out who is online and to check if they received a message. Finally, the user interface would be improved to make the design more pleasing and uniform.

Conclusion

The project has been completed meeting all the minimum requirement, with the user securely login to the server, the users can find other user on other computers, users can create and maintain a profile page and their can send and receive messages/files to online users. The application has been made with additional features which enhances the experience of using the application. A hybrid network has been used to develop the application.

Appendix


Login Server
HTML/CSS
CherryPy
Python
Database
APPLICATION (USER 1)
HTML/CSS
CherryPy
Python
Database
APPLICATION (USER 2)
HTML/CSS
CherryPy
Python
Database
APPLICATION (USER 3)

1.