



**Module Code & Module Title**

**CS6P05NI Final Year Project**

**Assessment Weightage & Type**

**25% FYP Interim Report Documentation**

**Semester**

**2021 Autumn**

**PROJECT TITLE: E-Kura Encrypted Chat Application**

**Student Name: Rishabh Karki**

**London Met ID: 20049209**

**College ID: NP01NT4S210042**

**Internal Supervisor: Aaditya Khwakhwali**

**External Supervisor: Bijay Limbu**

**Assignment Due Date: 28th December 2022**

**Assignment Submission Date: 28th December 2022**

**Word Count (Where Required):**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

Abstract

With this project report explains about the different aspects of the chat application with their features and development in the future. It explains how different aspects of chats application can be compromised and how much important the encryption is needed. And also how much it focuses on developing a system made for the people

Table of Contents

[1. Introduction 5](#_Toc123029610)

[1.1. About the Topic 5](#_Toc123029611)

[1.2. Problem Scenario 6](#_Toc123029612)

[1.3. The project as a solution 8](#_Toc123029613)

[1.4. Aims and Objective 9](#_Toc123029614)

[1.4.1. Objectives 9](#_Toc123029615)

[1.4.2. Structure of report 10](#_Toc123029616)

[Background 10](#_Toc123029617)

[Development 10](#_Toc123029618)

[Analysis of Project 10](#_Toc123029619)

[Future Work 10](#_Toc123029620)

[2. Background 11](#_Toc123029621)

[2.1. Understanding the project 11](#_Toc123029622)

[2.2. Similar Project Review 12](#_Toc123029623)

[2.3. Comparison Table 15](#_Toc123029624)

[2.4. Analysis and Conclusion of the Comparison 16](#_Toc123029625)

[3. Development 17](#_Toc123029626)

[3.1. Considered Methodology for the development 17](#_Toc123029627)

[Iterative Methodology 17](#_Toc123029628)

[Waterfall Methodology 18](#_Toc123029629)

[Selected Methodology 19](#_Toc123029630)

[Work Breakdown Structure 21](#_Toc123029631)

[4. Analysis of Process 23](#_Toc123029632)

[4.1. Progress Table 23](#_Toc123029633)

[4.2. Progress Review 23](#_Toc123029634)

[4.3. Progress Timeline 24](#_Toc123029635)

[4.4. Action Plan 24](#_Toc123029636)

[5. Future Work 24](#_Toc123029637)

[5.1. Phases to complete 24](#_Toc123029638)

[5.1.1. Web-development 24](#_Toc123029639)

[5.1.2. Integration and Testing 25](#_Toc123029640)

[5.1.3. Finalization and Deployment 25](#_Toc123029641)

[5.1.4. Documentation 25](#_Toc123029642)

[6. Conclusion 26](#_Toc123029643)

[7. References 27](#_Toc123029644)

[8.2. Appendix II (Resources Required) 32](#_Toc123029645)

# Table Of Figures

[Figure 1 chat application and privacy (Rakheja, 2021) 7](#_Toc123548514)

[Figure 2 User Interface of chat application 9](#_Toc123548515)

[Figure 3 Omegle Chat Logo (Wikipedia Commons, 2014) 13](#_Toc123548516)

[Figure 4 Chat Sansar Logo (Chat Sansar, 2022) 14](#_Toc123548517)

[Figure 5 KIK Logo (LOGOS, 2022) 15](#_Toc123548518)

[Figure 6 cycle of methodology (Art of Testing, 2021) 18](#_Toc123548519)

[Figure 7 steps of waterfall methodology (Adobe Communication , 2022) 19](#_Toc123548520)

[Figure 8 Work breakdown structure 22](#_Toc123548521)

[Figure 9 Gantt Chart 23](#_Toc123548522)

[Figure 10 Front end Page 29](#_Toc123548523)

[Figure 11 Verification Page during sign in 30](#_Toc123548524)

[Figure 12 Sign in Page 31](#_Toc123548525)

[Figure 13 Sign-up Page 32](#_Toc123548526)

# Introduction

## About the Topic

A software program known as a web-based chat application enables users to speak with one another in real time through a web browser. Users are able to communicate from any device with an internet connection because to the fact that these programs are often housed on a central server and accessed via a web browser.

The advent of web-based chat programs completely changed how people communicate online. Prior to the widespread acceptance of these programs, the main methods of communication were email and instant messaging programs that required installation on a particular device. The introduction of web-based chat made it simpler for users to keep in touch with friends and co-workers because they could access their conversations from any device that had a web browser.

Users could converse immediately through web-based chat programs rather of waiting for a reply to an email or instant message. As a result, communication was more effective and collaboration and real-time working were made simpler.

These days, web-based chat programs play a crucial role in how we communicate online. Both individuals and businesses use them for a variety of purposes, from socializing with friends to collaborating and communicating on the job.

## Problem Scenario



Figure chat application and privacy (Rakheja, 2021)

With the changing time messaging application is on the rise with different purpose and objective. But today’s technology lacks various important features like good encryption techniques and better integration. The major problem is How to offer security and verification for in-app transactions is one of the main difficulties messaging app firms confront. Is a basic password sufficient? Should there be two-factor authentication using an SMS text message-based one-time passcode (OTP)? In contrast to China, where those figures are 45% and 30%, 36% of Americans said they only use passwords, while just 11% said they use one-time passcodes delivered by SMS. The most popular alternative to utilizing passwords for verification in both nations is OTP SMS. (Shin, 2021)

The following are other factors of messaging applications they are as follows:

1. Lack of better integration

Elite firms' online chat apps have yet to succeed in providing better chat service integration, which results in a lot of frantic behaviour, lost network connectivity, and chats backup, which makes it difficult for users to converse.

1. Lack of better privacy

User privacy may be jeopardized by some chat applications that may gather and share user data with outside parties. Before using any chat program, it is crucial to thoroughly read the terms of service and privacy policies to be sure that your data will be treated in a way that you are comfortable with.

1. Lack of good user experience

Depending on the particular application, the user's device, and the internet connection, the user experience of web-based chat applications can vary significantly. Some chat programs could be challenging to use or function poorly, which could have a detrimental effect on the user's experience.

1. Spam problems and un-advantageous promotions

Due to how simple it is for users to transmit messages to huge audiences, web-based chat services can serve as a breeding ground for spam and false information. When receiving messages from unidentified sources, users should exercise caution and make sure the material is accurate before sharing it.

## The project as a solution

Graphical user interface, application

Description automatically generated with medium confidence

Figure User Interface of chat application

The lack of a solid user interface and many chat software' serious privacy and integration problems deter people from using them as their primary messaging platforms. The "E-Kura" project is a web-based program that offers a great user interface, a decent encryption approach for managing passwords and messages, and allows the user to integrate well with their chat application behavior.

The following web application uses bootstrap to provide a stunning user interface with realistic interactive behaviour. Additionally, PHP is a backend service for message encryption and decryption, OTP verification, and for improved data management XAMPP is used as the database storage.

This is how the chat application procedure works. A user is welcomed by a good user interface where he or she can sign up or create an account, along with the option to sign in using Google Authentication, making things simpler. He or she can converse concurrently with other members whose accounts were also created in this way. He is given access to a user list where he can browse the users on his list. The chat communication is encrypted using end-to-end encryption.

## Aims and Objective

The major aim of the following application is to provide privacy and a safe conversation environment for social people including good encryptions for OTP generation and messages.

### Objectives

It focuses on the following relative topics:

1. Provide a safe and private communication platform: Ensuring the security and privacy of communication is one of the key goals of an end-to-end chat web application. This includes preventing the messages from being viewed by anybody other than the intended receiver and preventing them from being intercepted or altered.
2. Real-time conversation facilitation: Enabling real-time communication between users is another goal of a chat online service. This can incorporate tools like group chat, and instant messaging.
3. Increase accessibility and reach: An end-to-end chat online application can attempt to increase accessibility and reach by supporting a variety of hardware and software, including desktop programs, mobile devices, and web browsers.
4. Enhance user experience: A flawless and delightful user experience is what an end-to-end chat web application aims to give to its users. This comprises attributes like an intuitive user interface, quick performance, and flexible design.

.

### Structure of report

### Background

The following report includes the pre-alpha development of the project which explains the front-end coding, its objective, and the basic structure of the working mechanism of the application, and outlines the future work and UI design of the application. This section includes research materials for comparable projects and a comparison table describing the project's salient characteristics and elements.

### Development

The following section includes pre-alpha development and explains the different stages of the development phases with the Gantt chart explaining the different time periods where the development process is taken. In a work breakdown structure, the phases of the project's development are divided into sections along with timelines.

### Analysis of Project

This section includes the developmental phase of the project explaining the research and development phases. In my individual case, it highlights the shortcomings events faced beyond the expectations present in Gantt's chart. And an immediate solution for project accomplishment comes with the time framework.

### Future Work

This is the last part of documentation where future work is included. This includes the future upcoming remaining development that needs to be done. Adaptation plans are also included in this section.

# Background

## Understanding the project

Project Elaboration

The following project uses bootstrap as a front-end and PHP as a backend. It includes a fantastic-looking User interface where a user can log in or signup for their account. There are set of user database which helps in storing and manipulating the data like message database where the messages of the user are encrypted and stored, user database where the information of the user is set and can be manipulated as per their requirements, and password encrypted database where the database of passwords are set. OTP (One-Time Password) is also generated through the backend which helps in the verification of the user. It is quite useful for resetting the password as well which alters the database.

Project Deliveries

The following chat application project is targeted at young people who want a new chat program for communication. It fulfils high demand and has a solid backend and database structure. Its primary goal is to deliver trustworthy encrypted chat services with secure database management. The application's user interface has been expertly designed to make it simple and easy for users to use. The project will also be planned such that it can operate with high availability, flexibility, and no buffers or stutters.

## Similar Project Review

**Project 1: Omegle web chat**

The website Omegle was created primarily to let users chat with random individuals. It functions by arbitrarily pairing users for one-on-one text or video conversations. while not having to register, users might move with strangers on the free online chat platform Omegle. Users area unit paired up in one-on-one chat sessions on the website wherever they will converse victimization the handles "You" associated "Stranger" in an anonymous manner. additionally, Omegle offers a operation referred to as "Spy Mode" that allows users to cause a matter to 2 random users and then listen in on their voice communication as a spectator. (D, 2021)



Figure Omegle Chat Logo (Wikipedia Commons, 2014)

Since its 2009 unleash, Omegle has grownup to be a popular platform for on-line chemical analysis and communication. It has, however, additionally come back vulnerable for lacking restraint, that has sometimes resulted in inappropriate behavior and bullying. Omegle has thus taken steps to do to extend the protection of its users, together with fixing a "moderated" portion of the web site wherever users will solely connect with different users UN agency are approved by Omegle's moderators and giving support for people UN agency have practised abuse. the web site Omegle was created primarily to let users chat with random individuals. It functions by arbitrarily pairing users for one-on-one text or video conversations. (Childnet, 2021).

**Project 2: Chat Sansar**

Online chat rooms can be found on the free chat website ChatSansar.com. A social chat network based on IRC is called ChatSansar. The translation of Chat Sansar into English is Chat World. A live virtual chat room where you can meet new people and make friends is called ChatSansar. Run by ChatSansar, chat rooms are available for Nepal, Pakistan, and India. Text chat by Mibbit and Video & Voice chat by KiwiIRC, both with facilities for sharing image and video files, are available on ChatSansar. With options like radio, voice calls, and video chat in our Voice & Video Chat option, chat rooms are much more enjoyable. A good chat room is CS.

(Chat Sansar, 2022)



Figure Chat Sansar Logo (Chat Sansar, 2022)

**Project 3: Kik**

With the help of the free messaging software Kik, users can communicate with one another and share texts, pictures, videos, and other types of media. The software may be downloaded for iOS, Android, and Windows Phone devices, and registration is not necessary to use it. Users can instead communicate with one another by sending messages to their username, which they create together with a password.



Figure KIK Logo (LOGOS, 2022)

Kik offers a variety of extra services in addition to chatting, such as the ability to start group chats, send stickers and GIFs, and take part in chat rooms. The "Kik Codes" feature of the app enables users to scan a special code to add new contacts or join groups. It has also drawn criticism for the lack of security and moderation, which has allowed for instances of bullying and the sharing of indecent content. As a response, Kik has taken steps to try to increase the safety of its users, including putting resources at the disposal of users who have suffered harassment and putting in place a system for reporting offensive information.

## Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Project 1 | Project 2 | Project 3 | My project |
| End to End Encryption | No | No | Yes | Yes |
| Chat History | No | No | No | Yes |
| Google Authentication Sign in | No | No | No | Yes |
| User account control | No | No | No | Yes |
| Easy to Use | No | No | Yes | Yes |
| Web-Application | Yes | Yes | No | Yes |
| Six-Digit OTP | No | No | No | Yes |
| Email verification | No | No | No | Yes |

## Analysis and Conclusion of the Comparison

When compared to mine, the next three initiatives are pretty comparable and move on with the same goal of communication. Both Chat Sansar and Omegle are web-based projects that are comparable to mine. In contrast to Omegle, Kik, and Chat Sansar, my project offers two-factor authentication, user control services, a superior user interface, and is simple to use. Additionally, it uses Google's API to enable users to sign in, simplifying the sign-in procedure. For the Chat UI's color filters and adjustments, I used Omegle and Chat Sansar as models.

# Development

## Considered Methodology for the development

### Iterative Methodology

As opposed to attempting to build the full system at once, iterative development includes building and distributing software in manageable chunks. Using this method, the project's overarching goals and objectives are first established before being divided into smaller components that may be produced and evaluated independently.

Diagram

Description automatically generated

Figure cycle of methodology (Art of Testing, 2021)

Some of the other steps are repeated after the initial requirement planning stage is finished. The final product is enhanced and iterated upon as these cycles are finished and put into action.

1. The planning stage is the initial step. It is employed to outline the specific needs. Whether it be software or hardware. Here, we also get ready for the subsequent steps.
2. The analysis stage is the second step. It is done to see if the project contains the necessary models and business logic or not.
3. The design phase follows. The project team should at this point have a complete list of requirements to work from, together with information on the project's direction and a conceptual systems design.
4. Implementation and coding make up the fourth stage. At this stage, all requirements, planning, and design plans are put into action and are coded. At this stage of the project, the system's real construction gets underway.
5. The testing stage is the sixth step. Here, several standards and norms are tested against the current build iteration to see if they are met. The purpose of these testing techniques is to identify any faults or errors in our system.
6. Eventually, when all of these steps have been finished. A thorough analysis of the system created up to this point is conducted. The system may be examined by the development team and stakeholders, who can also provide comments on various system components. (Art of Testing, 2021)

### Waterfall Methodology

A sequential procedure known as the "waterfall methodology" is used to manage projects. Progress is made step-by-step through the phases of conception, initiation, analysis, design, building, testing, deployment, and maintenance.

According to some, the Waterfall process adheres to the maxim "measure twice, cut once." The amount and calibre of work done upfront, including the documentation of all features, variations, user stories, and user interfaces, is what determines how well the Waterfall process works.

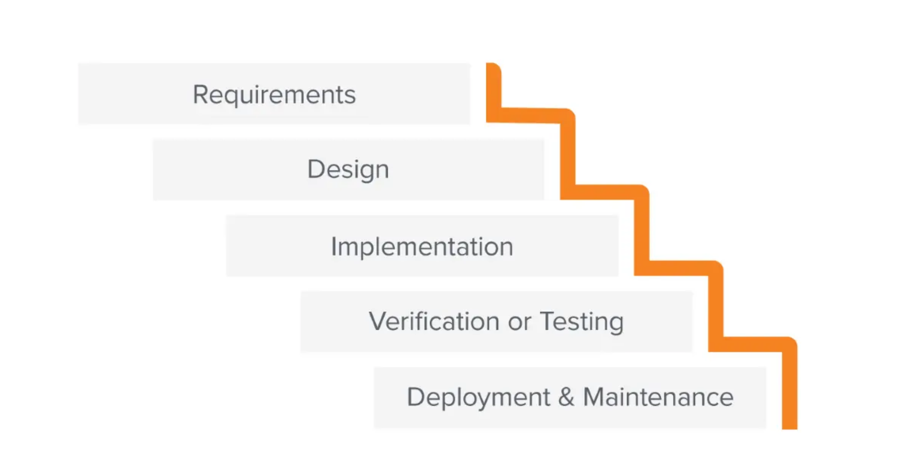


Figure steps of waterfall methodology (Adobe Communication , 2022)

Since the bulk of the research is done up front, more precise time estimates for each requirement can be made, which can lead to a more predictable release date. It is more difficult to shift direction with a Waterfall project than it is with Agile methodology if parameters change along the way.

The steps of the waterfall methodology are:-

1. A technological solution to the issues outlined in the product requirements is created by software developers, including scenarios, layouts, and data models.
2. The design is finished, and then technical implementation begins. Due to the meticulous study and design that has already been done, this phase of the Waterfall process may be the shortest. Programmers create programs during this phase based on project specifications, with some testing and implementation also occurring.
3. Testing is necessary before a product is made available to users to make sure there are no flaws and that all requirements have been met, resulting in a positive user experience with the program.
4. The maintenance phase starts after the program has been distributed to clients or deployed in the market. A team will be assigned to handle upgrades and the release of new software versions as bugs are discovered and user change requests are received. (Adobe Communication , 2022)

### Selected Methodology

For this very project, I choose an evolutionary method. A technique to software development called evolutionary development, also known as iterative and incremental development, entails creating and releasing minor, incremental updates to a product over time. This method is predicated on the notion that software may be created and delivered over the course of several iterations, with each iteration building on the work of the preceding one.

According to the evolutionary approach, work should be prioritized, divided into smaller portions, and then sent to the client one at a time. Both the number of chunks and the number of delivery to the consumer are enormous. The key benefit is that the customer's confidence grows since he continuously receives quantifiable goods or services to validate and verify his criteria from the start of the project. (Geeks for geeks, 2022)

The major positive aspects of the following methodology are:-

1. Delivery of useable software occurs sooner with the evolutionary development technique than it would if the full product were built at once because the product is created and delivered in a succession of tiny, incremental versions.
2. Greater flexibility and adaptability: Since the product is created over a number of iterations, it is simpler to update it as needed and adapt to brand-new requirements. When the project scope is unclear or is anticipated to alter dramatically, this can be especially helpful.
3. Better risk management: By dividing the project into smaller, incremental iterations, issues may be more easily identified and dealt with as they materialize. This can lessen the effect that any issues could have on the project as a whole.
4. Increased stakeholder engagement: The evolutionary development approach enables constant dialogue and collaboration with stakeholders, which can assist guarantee that the final product satisfies their requirements and expectations.
5. Enhanced learning and continuous improvement: The team can learn from its mistakes and make adjustments for the following iteration by reviewing the product at the end of each iteration. Over time, this can result in a more fruitful and productive development process

### Work Breakdown Structure

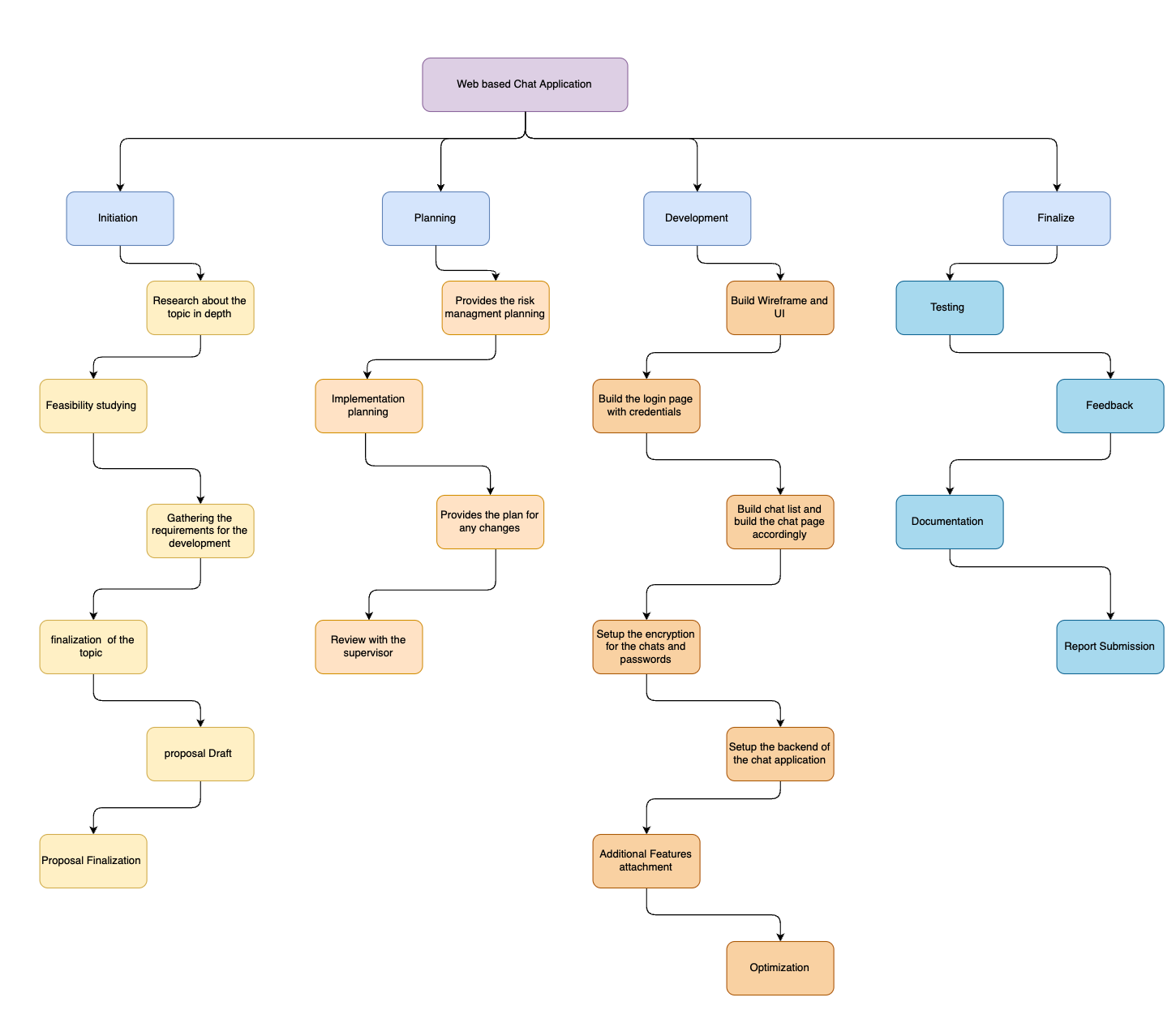


Figure Work breakdown structure

* 1. Gantt Chart

Chart

Description automatically generated

Figure Gantt Chart

# Analysis of Process

## Progress Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Task Name | Start Date | End Date | Status |
|  | Project Start Up | 12-Oct-2022 | 12-Nov-2022 | Completed |
|  | Project Draft | 20-Nov-2022 | 27-Nov-2022 | Completed |
|  | Project Submission | 27-Nov-2022 | 28-Nov-2022 | Completed |
|  | Requirement Finalization | 28-Nov-2022 | 7-Dec-2022 | Completed |
|  | UX/UI design | 10-Dec-2022 | 17-Dec-2022 | Completed |
|  | Web Design | 25-Dec-2023 | 20-Jan-2023 | Ongoing |
|  | Database Design | 20-Jan-2023 | 19-Feb-2023 | Not Started |
|  | Backend Development | 19-Feb-2023 | 14-Mar-2023 | Not Started |
|  | Testing | 15-Mar-2023 | 2-Apr-2023 | Not Started |
|  | Local Server test | 2-Apr-2023 | 4-Apr-2023 | Not Started |
|  | Deployment | 4-Apr-2023 | 6-Apr-2023 | Not Started |
|  | Documentation | 5-Apr-2023 | 17-Apr-2023 | Not Started |
|  | Submission | 17-Apr-2023 | 19-Apr-2023 | Not Started |

## Progress Review

Project Plan, Design, and Requirements

The methodological process is viewed as an evolutionary procedure in which a suitable division is made between development and documentation. The entire project's development, including its timing, is tracked using the Gantt Chart. The very first stage of development involves selecting a topic and initiating research and study. The topic is chosen, and the research is completed within a predetermined timeframe. Using bootstrap, the front-end code is created and made responsive. The design's UI/UX is heavily taken into account. It is a very user-defined and user-friendly application since a survey is conducted with twenty-one people's opinions for better assistance and clarity.

The required resources are explained in the Appendix

## Progress Timeline

The Gantt chart and milestone events are used to set the development process. UX design and some front-end code were completed prior to the Gantt chart. And according to it, a lot of work still needs to be completed, including the implementation of JavaScript and the design of the backend and databases. Additionally, the Google authentication sign-in process has not yet been implemented. Double-factor authentication and OTP generation encryption methods are being researched. The chat application's backend needs to be researched, and for better support, either Node.js or PHP could be used in the design.

## Action Plan

The tasks for the project will be finished according to the updated Gantt plan made for the interim report. The requirements, design, and development phases were completed in accordance with the methodology chosen and the Gantt chart. The construction work is finished. The following phases will be finished by the time this interim report is turned in.

# Future Work

## Phases to complete

### Web-development

**Front-end**

There will be some changes to the pages as JavaScript will be added with proper validation now that the majority of the UI/UX and front-end work has been finished. To simulate the loading of the database into the system, JavaScript will be used on the very first page of the loading screen. The JavaScript and Bootstrap code for the logged-in home screen, which includes the navbar and the user's profile picture, is still being written. When a message is delivered or received, ajax will be used in the project to refresh the page.

**Backend**

It has not yet been decided whether to use PHP or node.js to support and alter the database while working on the backend. The project also intends to implement features like conversation deletion and chat room organization by user name and recent history. The backend, which encrypts passwords and messages, also encrypts the database. It is also intended to upgrade the system to include features like OTP, two-factor authentication, and password reset capabilities. Additionally, Google authentication sign-in has yet to be enabled for user convenience.

### Integration and Testing

In this second stage of development, testing and integration are completed. The system's database is verified twice during this phase, and black box and white box testing are conducted to look for vulnerabilities. The project is then hosted locally so that it can be verified whether the message was delivered or not. Along with it, the admin page, where the user database and their preferences are kept, is controlled.

### Finalization and Deployment

In this final stage of development, Amazon services are used to launch the product onto the internet. The project is now complete, has all the settings, and is prepared for use by the general public. The beta website is then hosted and made available to the general public.

### Documentation

On this theoretical page of the project, all the options, settings, and environments are described along with a representation of the project. A prototype is displayed. Also finished at this stage of the project's development is the final documentation. The project's final report will then be finished and sent.

# Conclusion

The following project is overall quite user-friendly and well-designed. The project's main goal is to offer every function a chat application can have. The project's milestone will be reached on schedule, even though the report that follows indicates the alpha development.

# References

Shin, J., 2021. *4 Challenges Facing the Messaging Industry.* [Online]   
Available at: https://www.tyntec.com/blog/4-challenges-facing-messaging-industry

D, I., 2021. *Everything parents need to know about omegle.* [Online]   
Available at: https://www.mobicip.com/blog/everything-parents-need-know-about-omegle  
[Accessed 25 December 2022].

Childnet, 2021. *Key things parents and carers need to know.* [Online]   
Available at: https://www.childnet.com/blog/what-is-omegle-key-things-parents-and-carers-need-to-know/  
[Accessed 25th December 2022].

Wikipedia Commons, 2014. *wikipedia common.* [Online]   
Available at: https://commons.wikimedia.org/wiki/File:Logo\_Omegle.svg  
[Accessed 25th December 2022].

Chat Sansar, 2022. *Chat Sansar.* [Online]   
Available at: https://chatsansar.com  
[Accessed 25 December 2022].

Chat Sansar, 2022. *Chat Sansar.* [Online]   
Available at: https://chatsansar.com  
[Accessed 25th December 2022].

LOGOS, 1., 2022. *KIK MESSANGER LOGO.* [Online]   
Available at: https://1000logos.net/kik-messendger-logo/  
[Accessed 25th December 2022].

Art of Testing, 2021. *Iterative Model in Software Engineering.* [Online]   
Available at: https://artoftesting.com/iterative-model  
[Accessed 25th December 2022].

Adobe Communication , 2022. *Waterfall Methodology.* [Online]   
Available at: https://business.adobe.com/blog/basics/waterfall  
[Accessed 25th December 2022].

TotalRide , 2021. *Evolutionary Process Models in Software Engineering.* [Online]   
Available at: https://www.tutorialride.com/software-engineering/evolutionary-process-models-in-software-engineering.htm  
[Accessed 25th December 2022].

Geeks for geeks, 2022. *Software Engineering | Evolutionary Model.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-evolutionary-model/  
[Accessed 25th December 2022].

1. **Appendix**

## Appendix 1(Development)

**Front-end Development**

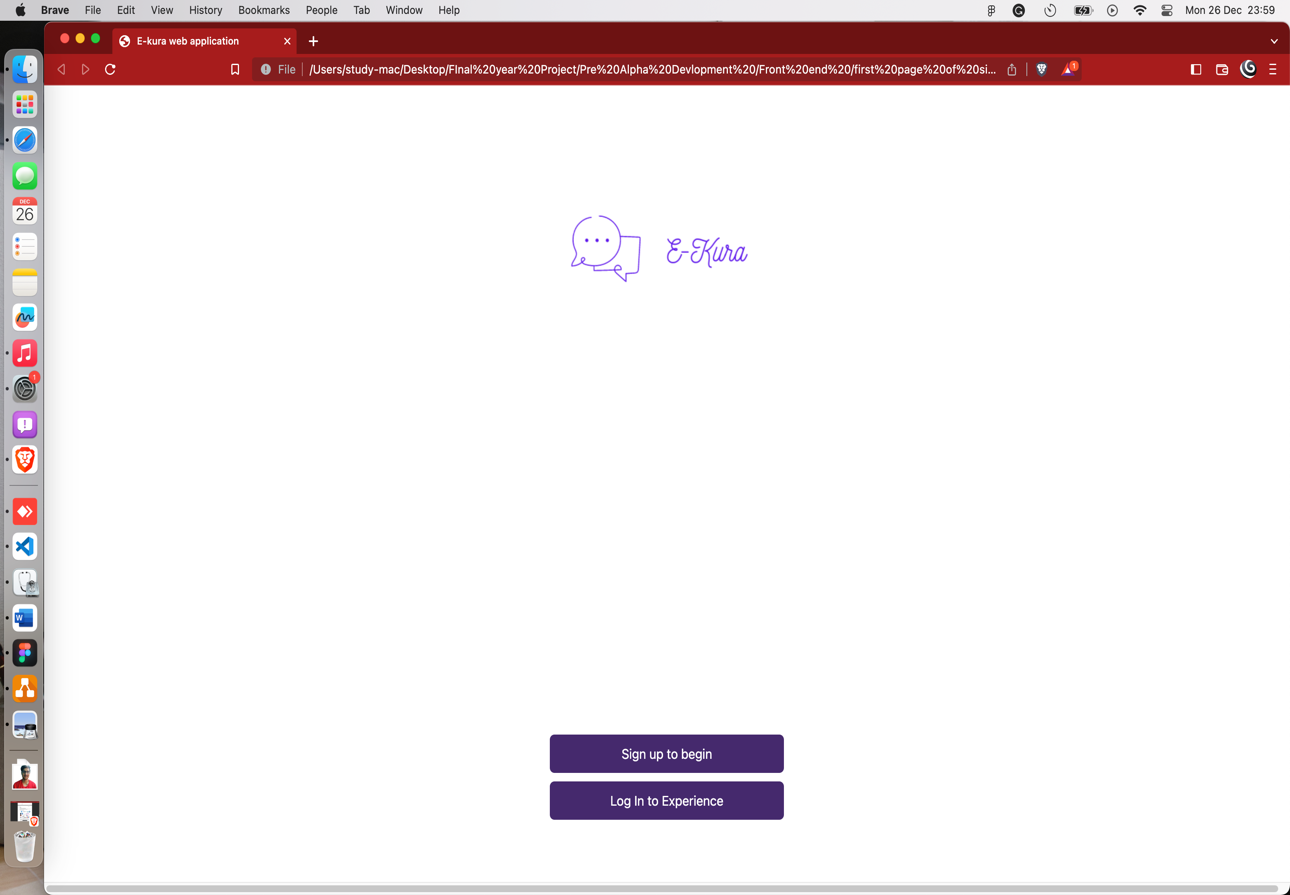
****

Figure Front end Page

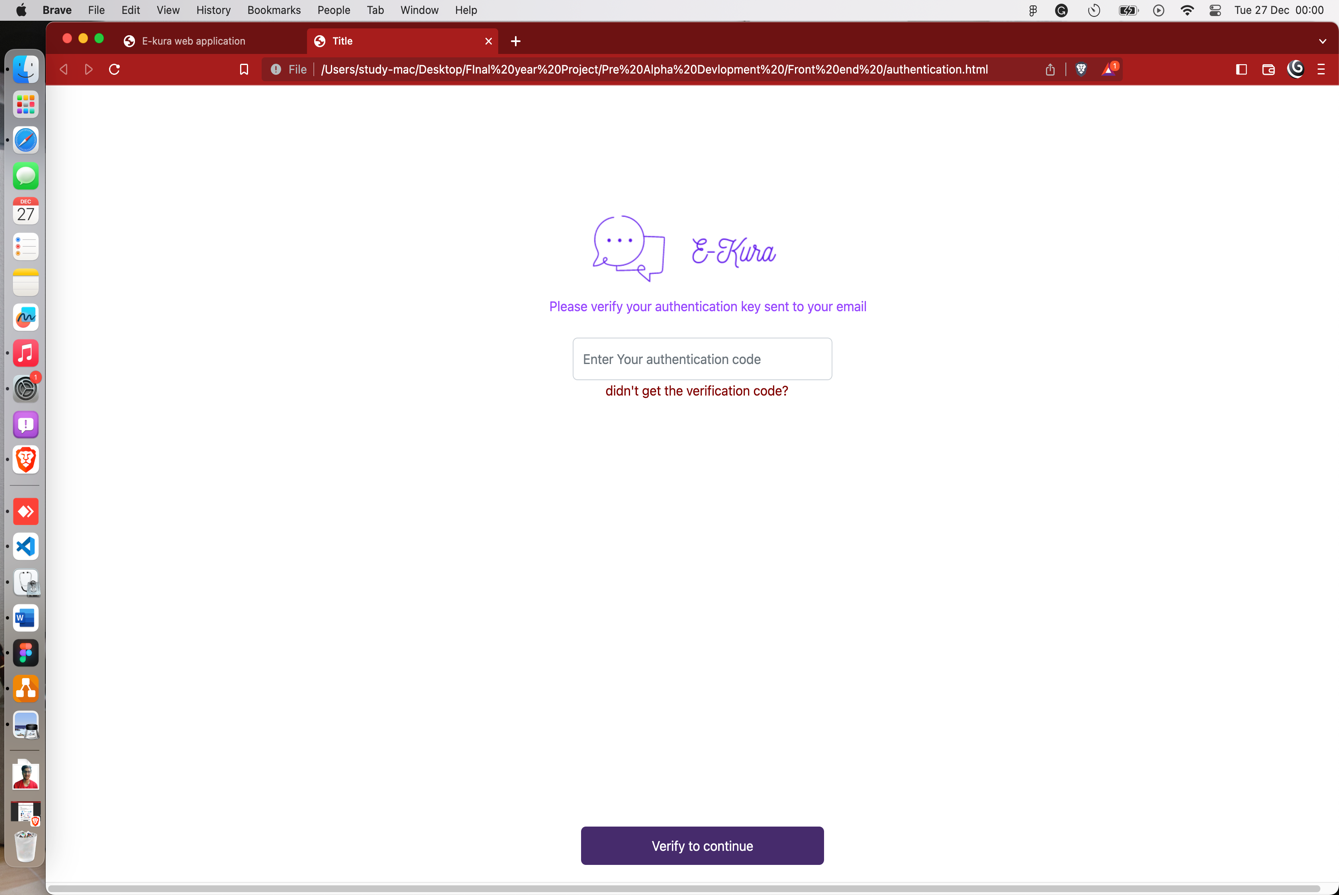
****

Figure Verification Page during sign in

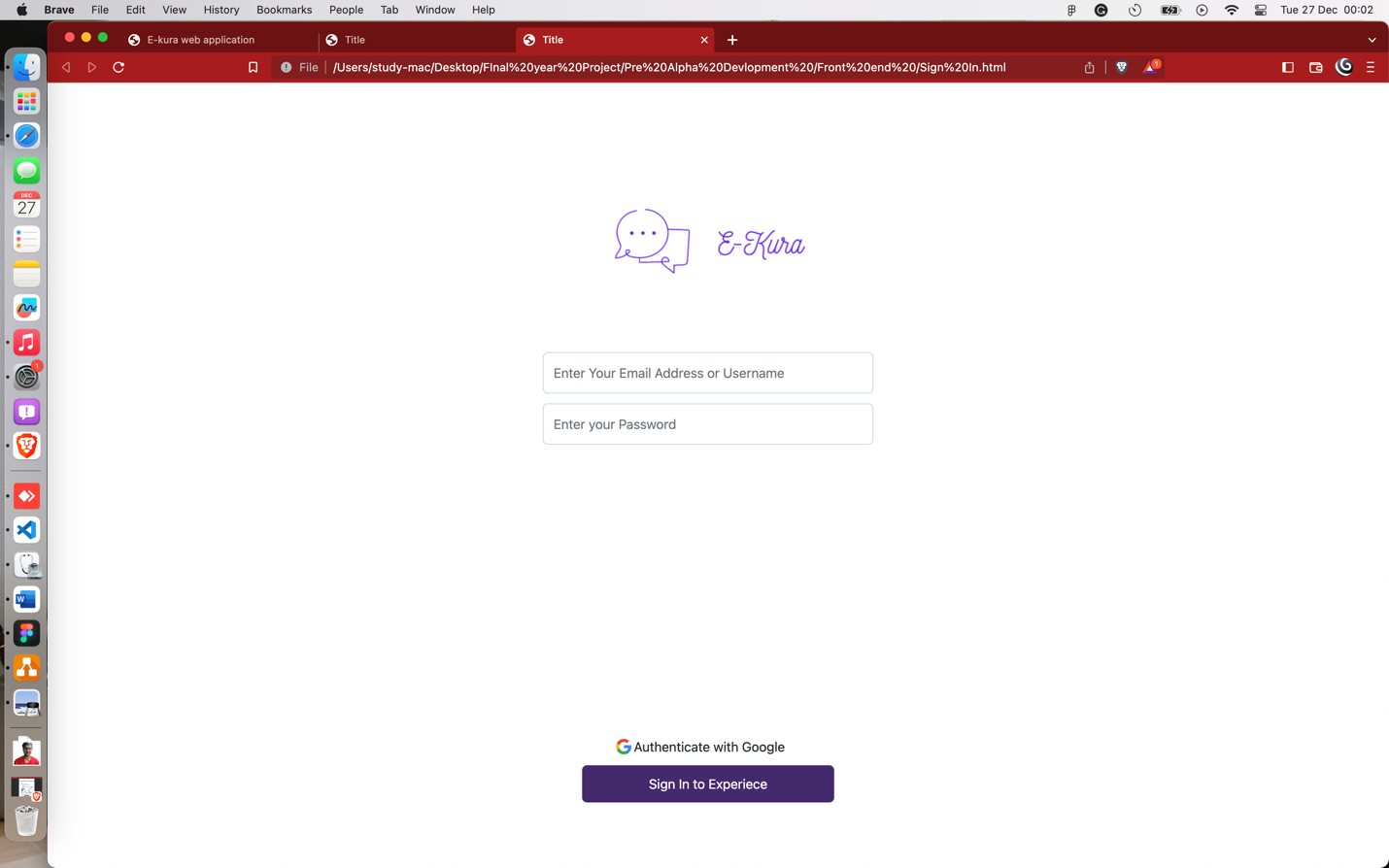
****

Figure Sign in Page

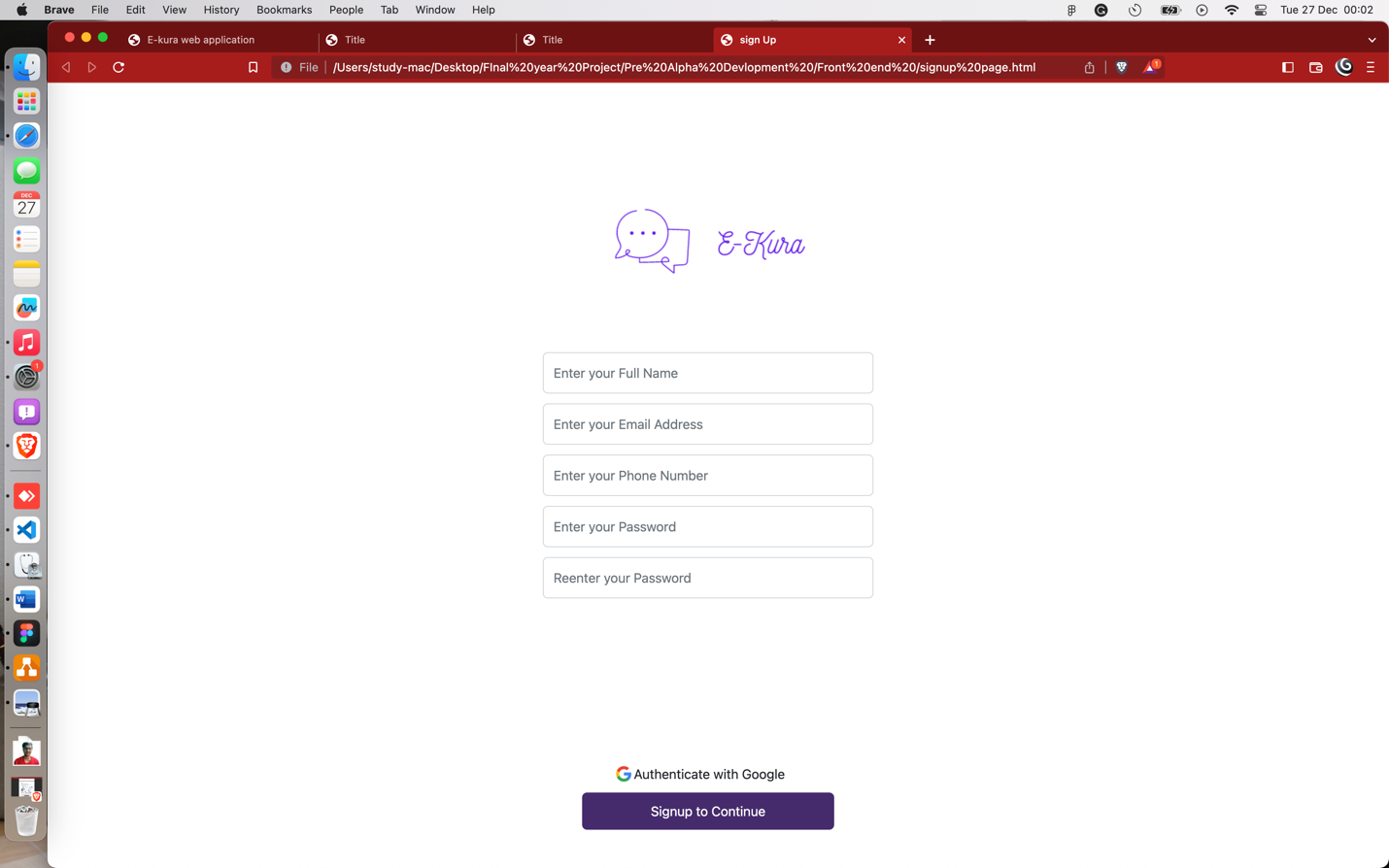


Figure Sign-up Page

## Appendix II (Resources Required)

For this project, software requirements that are needed for the development are listed below:



1. Figma

Figma is a cloud-based design and prototyping tool that enables teams to collaborate in real time on the design and development of websites, mobile apps, and other digital products. It offers a range of features including vector editing, prototyping, design components, and design handoff, making it a comprehensive solution for teams working on digital projects.

1. Bootstrap

Bootstrap is a front-end web development framework that provides pre-designed and pre-styled HTML, CSS, and JavaScript components for building responsive and mobile-first websites. It allows developers to quickly create professional and consistent-looking layouts by using pre-defined classes, rather than having to write custom CSS and layout code from scratch.

1. JavaScript

JavaScript is a programming language that is commonly used to add interactivity to web pages. It is a client-side language, which means it is executed by the user's web browser rather than on a server.

1. PHP

PHP is a popular programming language for web development, particularly for creating server-side scripts that interact with databases and generate dynamic web pages. It is known for its simplicity and flexibility and is widely used in combination with HTML, CSS, and JavaScript to build dynamic, interactive websites.

1. Laravel

Laravel is a popular open-source PHP framework used for web application development. It is known for its elegant syntax, robust tools for routing, authentication, and caching, and support for databases and servers.

1. Visual Studio Code

Visual Studio Code is a popular, cross-platform code editor developed by Microsoft. It includes features such as syntax highlighting, debugging, and version control integration, making it a powerful tool for developers working in a variety of programming languages.

1. XAMPP

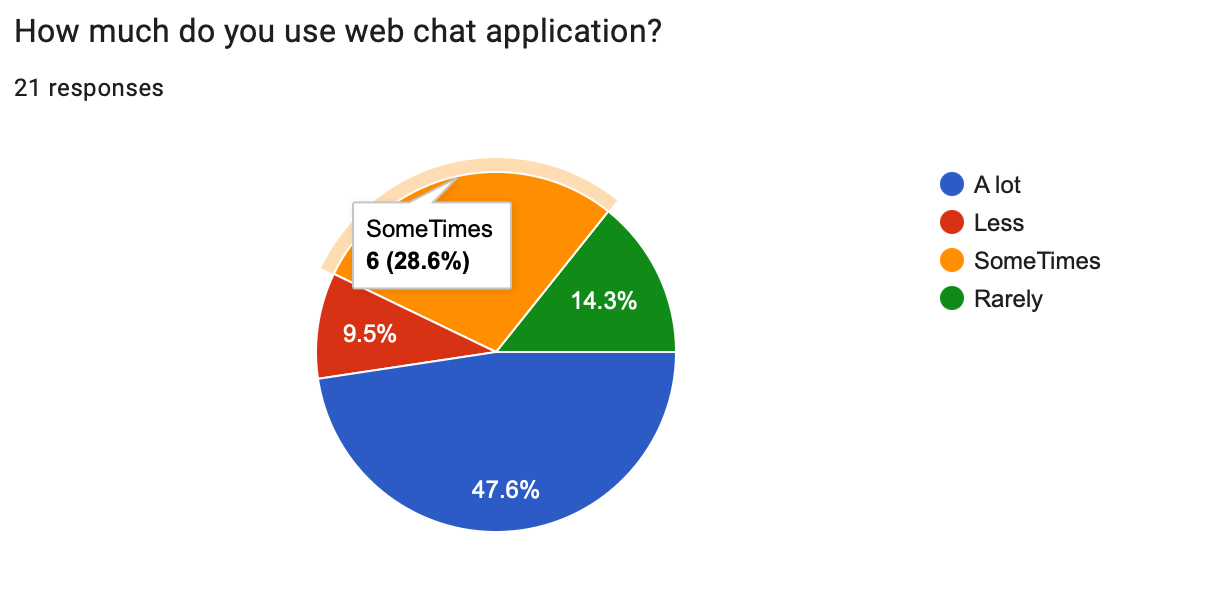
XAMPP is a free, open-source software stack that includes Apache HTTP Server, MySQL database, and PHP and Perl programming languages. It is designed to make it easy for developers to set up a local web server for testing and development purposes.

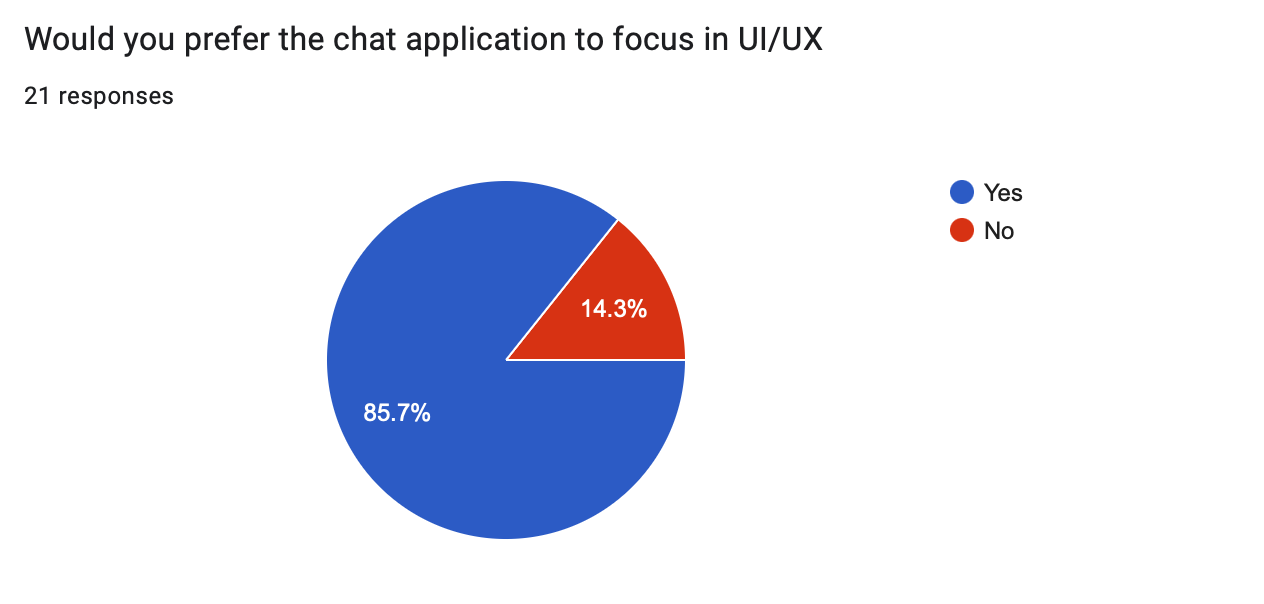
1. Node.js

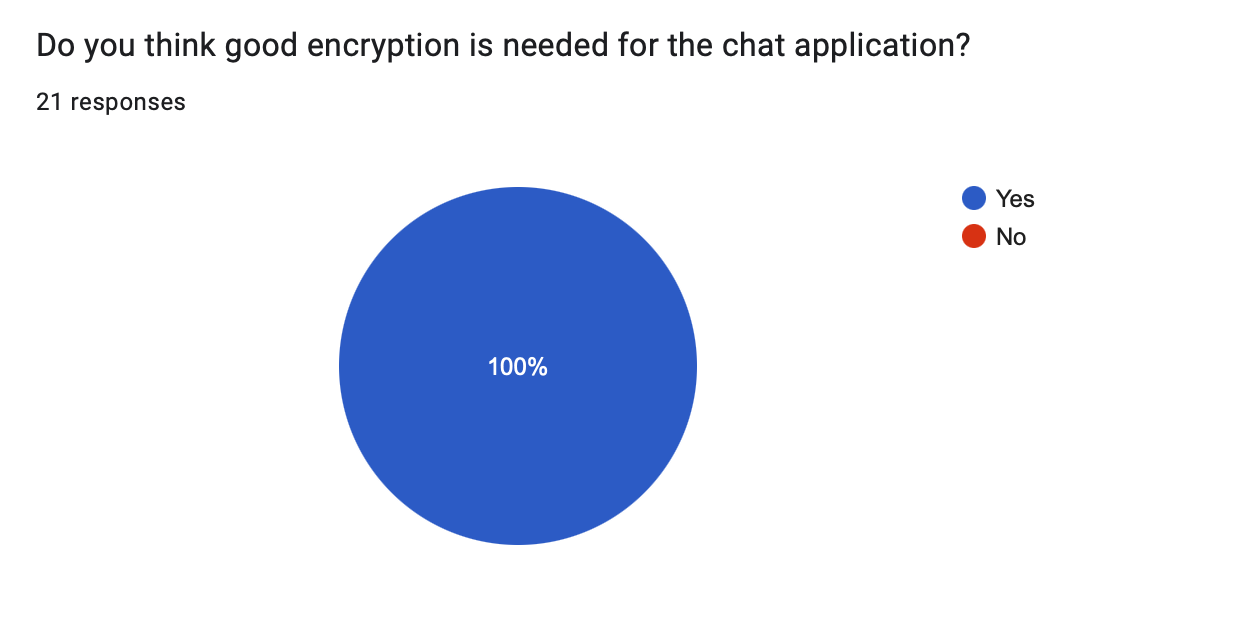
Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript on the server, making it possible to build fast, scalable network applications with JavaScript.

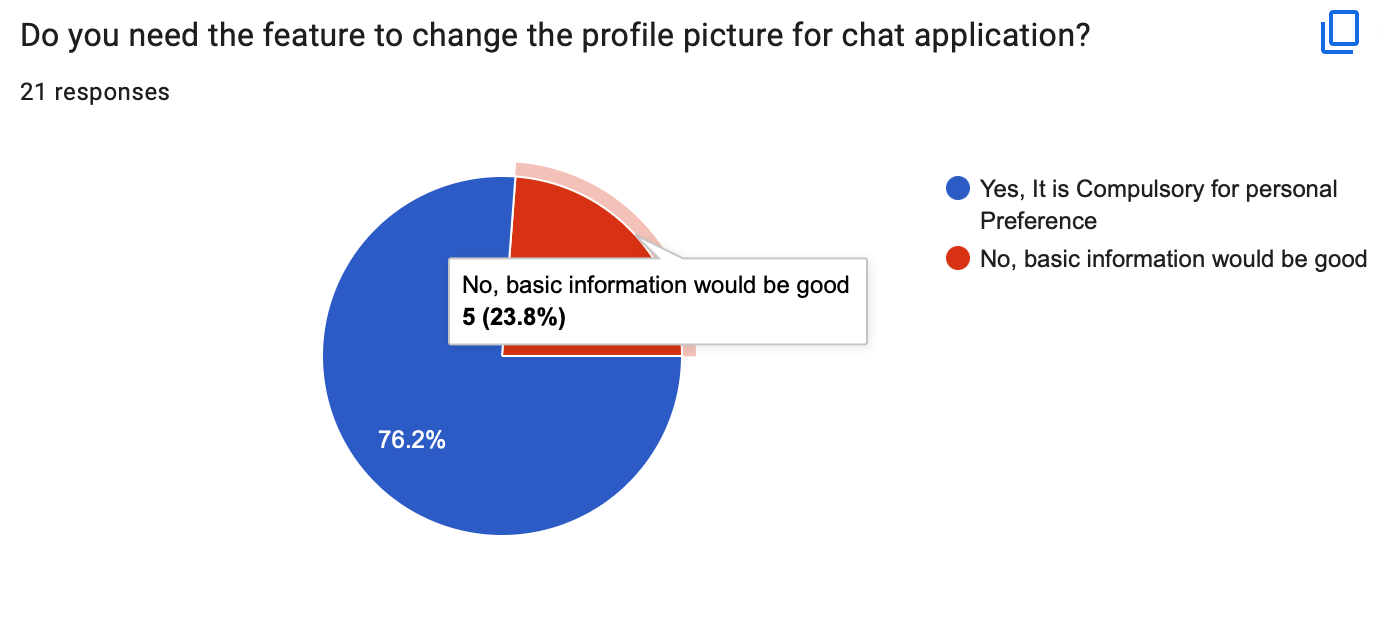
## Appendix III (Survey Questions)

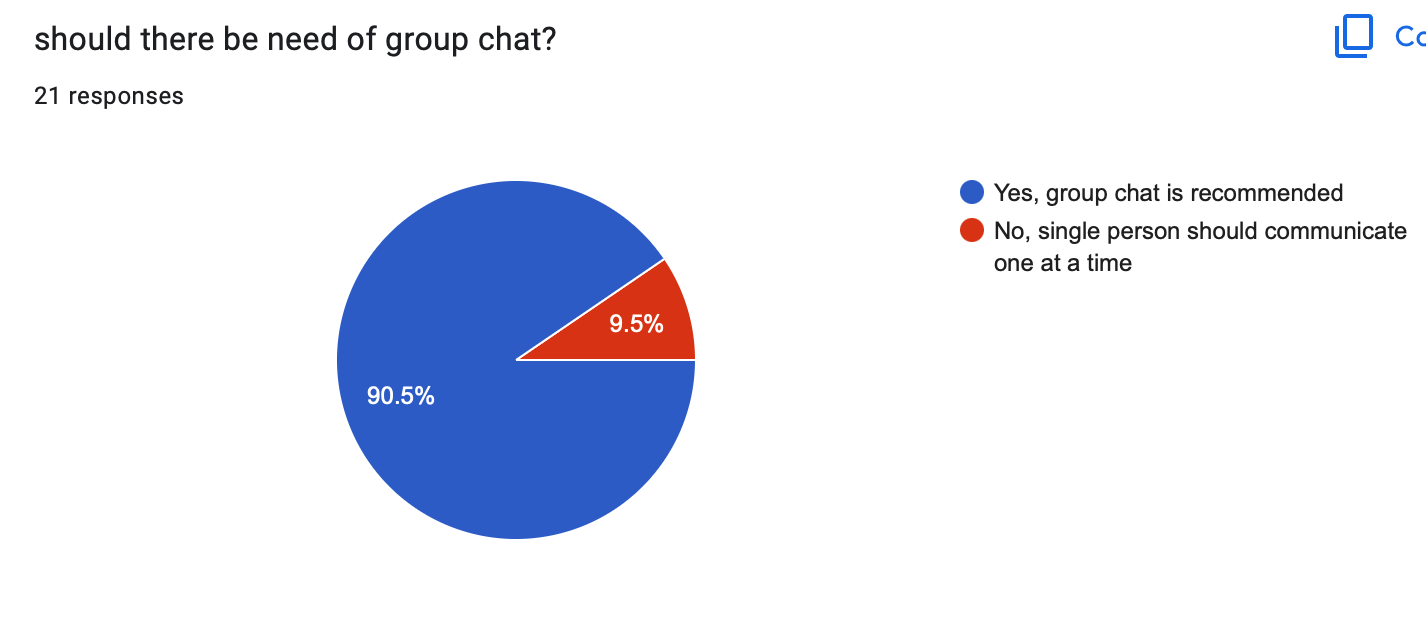
The survey was done on 21 people and their response are shown as:











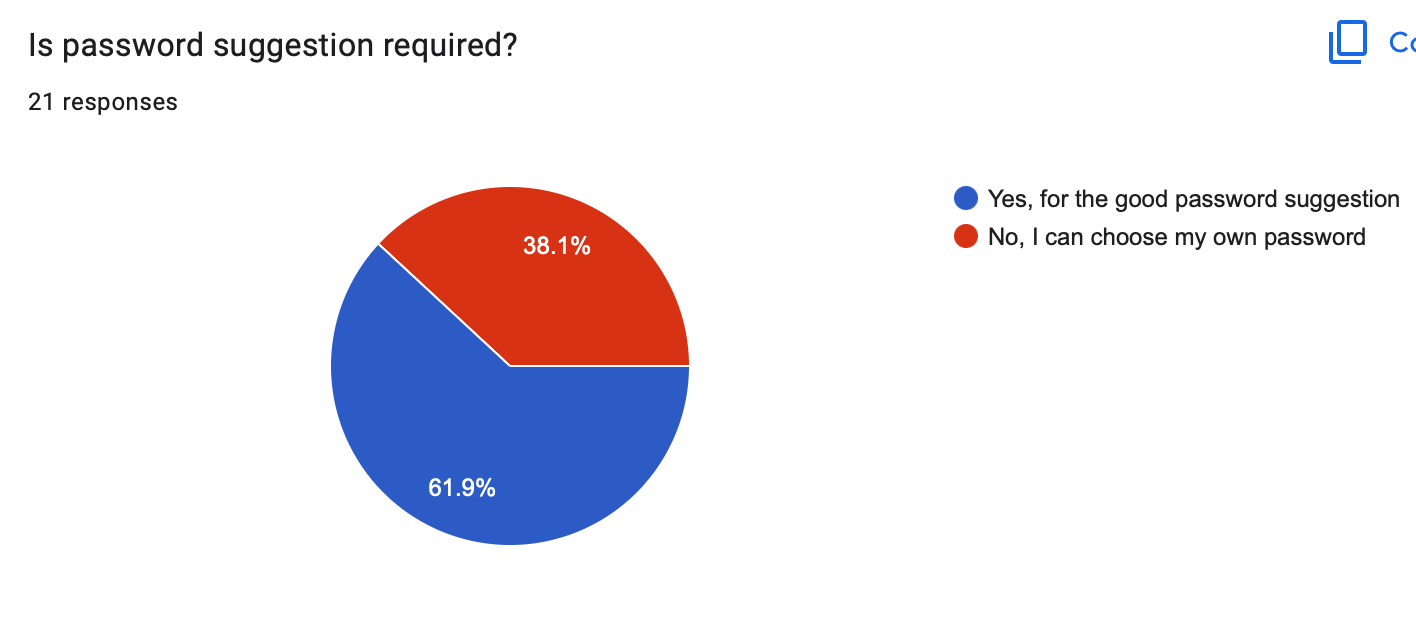
Chart, pie chart

Description automatically generated

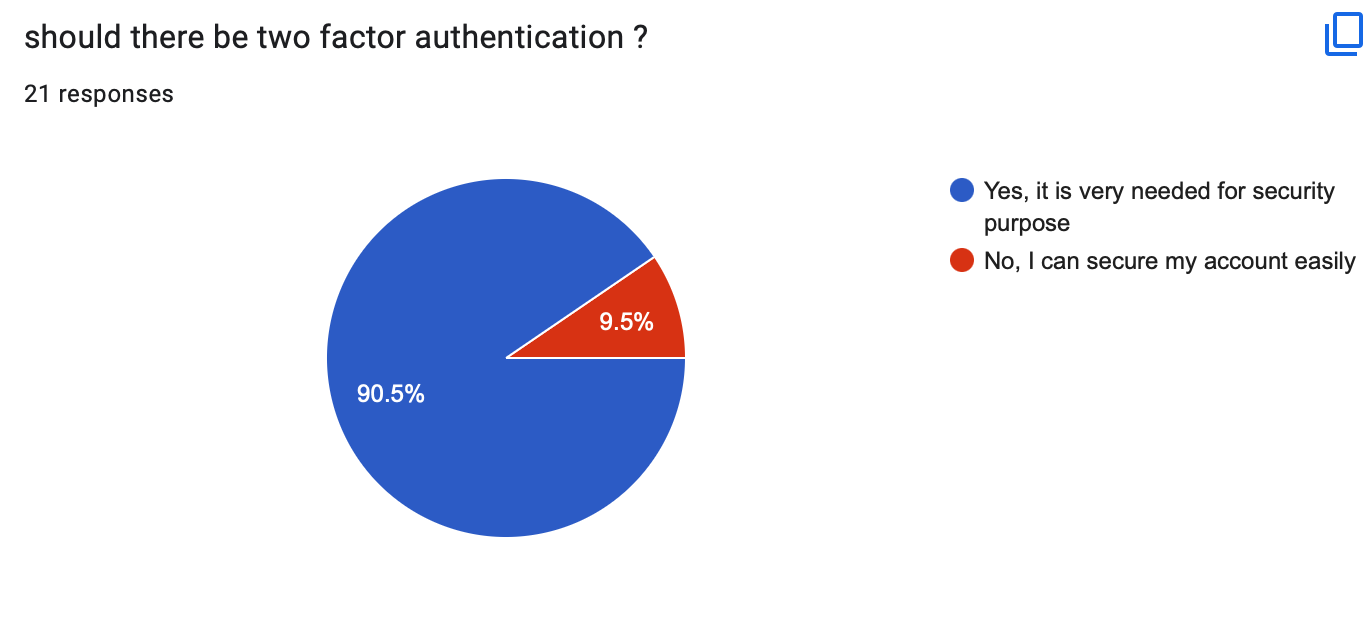
Chart, pie chart

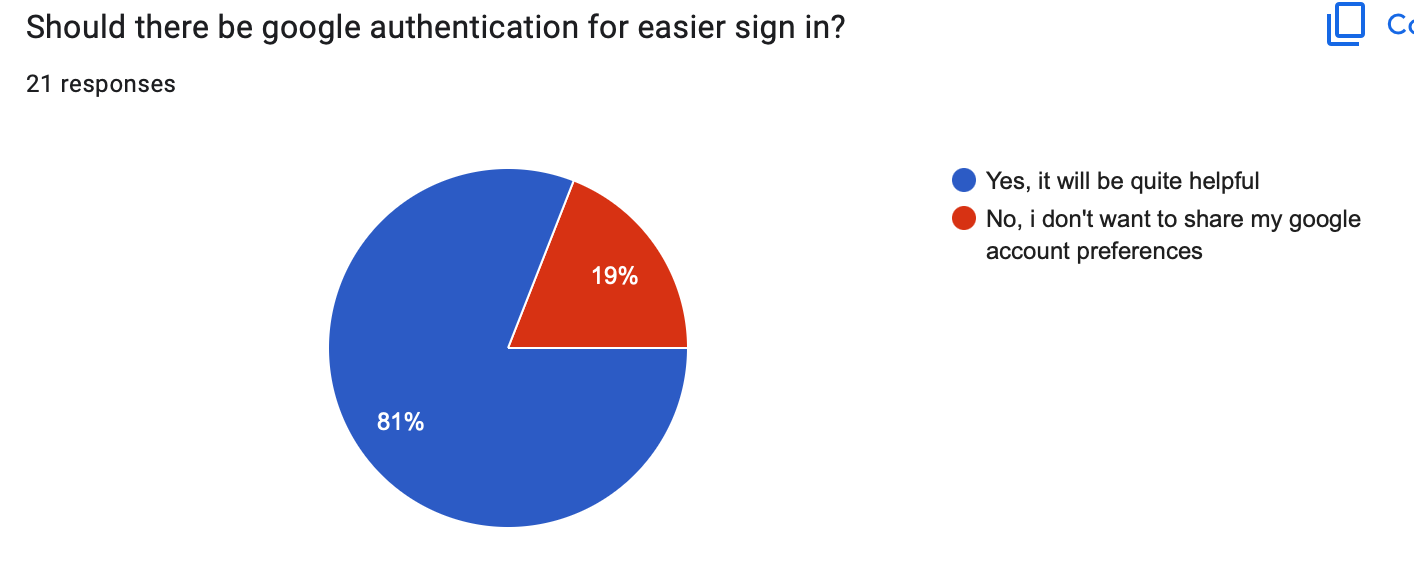
Description automatically generatedChart, pie chart

Description automatically generated



Chart, pie chart

Description automatically generated



## The majority of individuals use web chat applications, and according to the survey question that was collected, most of them claim to adhere to UX/UI guidelines for better UI. People need the flexibility to change their profile pictures in order to use excellent encryption in their own way. They require both the group chat and the visible messages features at the same time. Most of them need message encryption with password suggestions since they are so concerned about security and privacy. They also advised concentrating on two-factor authentication for simpler sign-in using the Google api.

## Appendix IV(Code)

For the first Page

<!doctype html>

<html lang="en">

<head>

<title>Title</title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS v5.2.1 -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<!-- place navbar here -->

</header>

<main>

<div>

<div class="d-flex align-items-center justify-content-center" style="height: 300px;">

<img src="pics/E-Kura 1.png" alt="your-image-description" style="height: 400px;">

</div>

</div>

<div class="container-fluid" id="blank">

</div>

<div class="containerfluid">

<div class="d-flex align-items-center" >

<a href="#" class="btn btn-primary mx-auto" id="signUp">Sign up to begin</a>

</div>

<div class="d-flex align-items-center" >

<a href="#" class="btn btn-primary mx-auto" id="signUp">Login to Experience</a>

</div>

</div>

</main>

<footer>

<!-- place footer here -->

</footer>

<!-- Bootstrap JavaScript Libraries -->

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"

integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous">

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"

integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" crossorigin="anonymous">

</script>

</body>

</html>

For the Authentication Page

<!doctype html>

<html lang="en">

<head>

<title>Title</title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS v5.2.1 -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<!-- place navbar here -->

</header>

<main>

<div class="container-fluid" id="blank3">

</div>

<div class="d-flex align-items-center justify-content-center" style="height: 200px;">

<img src="pics/E-Kura 2.png" alt="your-image-description" style="height: 125px;">

</div>

<div class="d-flex justify-content-center" id="verify">

<label id="verify">Please verify your authentication key sent to your email</label>

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield1" placeholder="Enter Your Verification Code">

</div>

<div class="d-flex justify-content-center">

<a href="#" class="text-decoration-none" id="verify">Didn't get the code?</a>

</div>

<div class="container-fluid" id="blank2">

</div>

<div class="d-flex align-items-center">

<a href="#" class="btn btn-primary mx-auto" id="signUp">Login to Experience</a>

</div>

</main>

<footer>

<!-- place footer here -->

</footer>

<!-- Bootstrap JavaScript Libraries -->

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"

integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous">

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"

integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" crossorigin="anonymous">

</script>

</body>

</html>

For the Sign In page

<!doctype html>

<html lang="en">

<head>

<title>Title</title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS v5.2.1 -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<!-- place navbar here -->

</header>

<main>

<div class="d-flex align-items-center justify-content-center" style="height: 200px;">

<img src="pics/E-Kura 2.png" alt="your-image-description" style="height: 125px;">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield1" placeholder="Enter Your Email Address">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield2" placeholder="Enter Your Password">

</div>

<div class="container-fluid" id="blank1">

</div>

<div class="d-flex align-items-center">

<a href="#" class="btn mx-auto" id="google"><img src="pics/g.png" height="40px" id="google"> Authenticate with Google</a>

</div>

<div class="d-flex align-items-center">

<a href="#" class="btn btn-primary mx-auto" id="signUp">Login to Experience</a>

</div>

</main>

<footer>

<!-- place footer here -->

</footer>

<!-- Bootstrap JavaScript Libraries -->

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"

integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous">

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"

integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" crossorigin="anonymous">

</script>

</body>

</html>

For the Signup Page

<!doctype html>

<html lang="en">

<head>

<title>Title</title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS v5.2.1 -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<!-- place navbar here -->

</header>

<main>

<div class="d-flex align-items-center justify-content-center" style="height: 200px;">

<img src="pics/E-Kura 2.png" alt="your-image-description" style="height: 125px;">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield1" placeholder="Enter Your Full Name">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield2" placeholder="Enter Your Email Address">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield2" placeholder="Enter Your Phone Number">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield2" placeholder="Enter Your Password">

</div>

<div class="d-flex justify-content-center">

<input type="text" class="form-control" id="textfield2" placeholder="Reenter Your Password">

</div>

<div class="container-fluid" id="blank">

</div>

<div class="d-flex align-items-center">

<a href="#" class="btn mx-auto" id="google"><img src="pics/g.png" height="40px" id="google"> Authenticate with Google</a>

</div>

<div class="d-flex align-items-center">

<a href="#" class="btn btn-primary mx-auto" id="signUp">Sign Up to Continue</a>

</div>

</main>

<footer>

<!-- place footer here -->

</footer>

<!-- Bootstrap JavaScript Libraries -->

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"

integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous">

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"

integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" crossorigin="anonymous">

</script>

</body>

</html>

For the CSS of the system

#blank{

height: 380px;

background-color: white;

}

#signUp{

background-color: #452A6D;

margin-bottom: 15px;

height: 45px;

width: 305px;

padding-top: 10px;

}

#Signin{

width: 305px;

}

#textfield1{

width: 414px;

height: 35px;

display: flex;

align-items: center;

justify-content: center;

}

#textfield2{

width: 414px;

height: 35px;

display: flex;

align-items: center;

justify-content: center;

margin-top: 10px;

}

#google{

height: 35px;

display: flex;

align-items: center;

justify-content: center;

border-radius: 2px;

margin-right: 4px;

}

#blank1{

height: 480px;

background-color: white;

}

#blank2{

height: 36px;

background-color: white;

}

#blank3{

height: 100px;

background-color: white;

}

#verify{

margin-bottom: 4px;

color: #452A6D;

}