

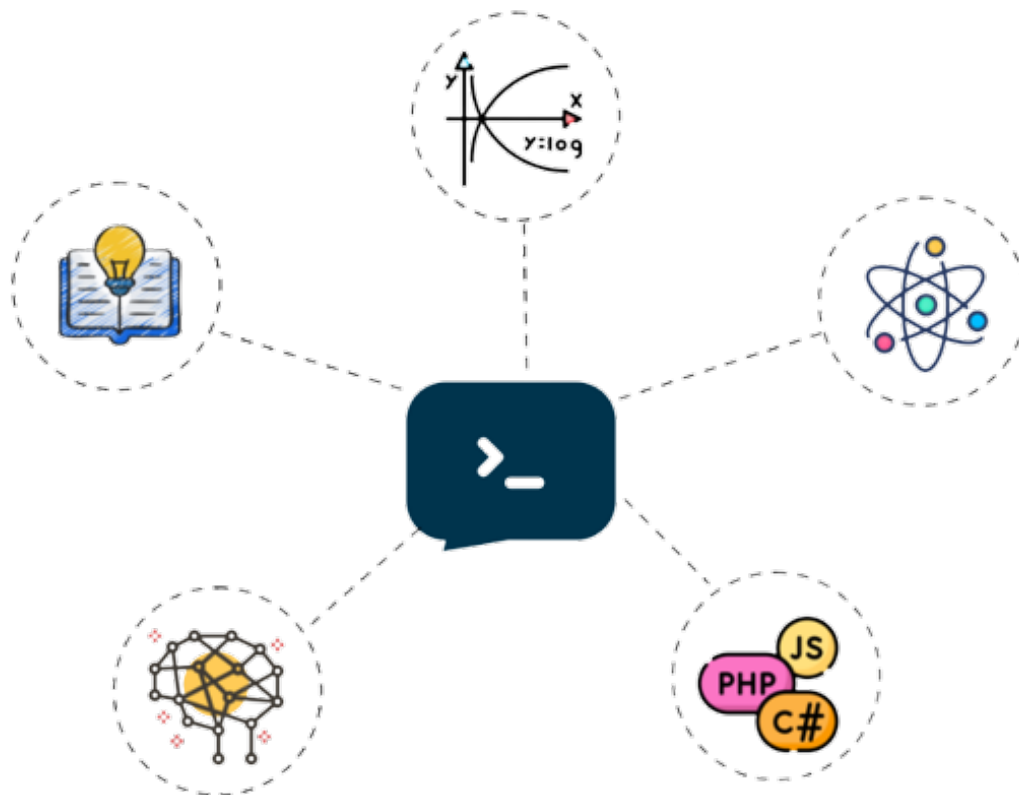


Cape Peninsula
University of Technology

**INDUSTRIAL COMPUTING DESIGN PROJECT 3
(DES371S)**

**Code Chat A Hypertext Instant Messaging Web-App
Mohlomi Cliff Makhetha**

**Student: 220118019
Due: 27 October 2023**



Bachelor of Engineering Technology in Computer Engineering
Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering and Built Environment

Contents

1	Introduction	4
2	Problem Description	4
3	Literature Review	5
3.1	Rich Text Formatting:	5
3.2	Integration of Third-Party Services and Functionalities	5
3.3	Educational Learning Management Systems (LMS):	7
4	Objectives of the Project	8
5	Conceptual Designz	10
6	Methodology	11
7	Project Deliverables	13
8	Project Plan	14
9	Project Budget	16
10	ECSA Graduate Attributes	17
10.1	Graduate Attribute 1 Problem Solving	17
10.2	Graduate Attribute 3 Engineering Design	17
11	Appendix	20

List of Figures

1	Jira's Markdown Formatting Textarea	5
2	Jira's Markdown Formatting Textarea	6
3	Jira's Markdown Formatting Textarea	6
4	Jira's Markdown Formatting Textarea	7
5	Jira's Markdown Formatting Textarea	7
6	Embedding of Syntaxes like Latex, Markdown, Vexflow, and Smiles	8
7	Time and Project Management tools	9
8	Conceptual Plug In Design of the Code Chat System Solution	10
9	Project Schedule Gantt Chart	14
10	Front Page for Joining a Group Chat	20
11	Front Page for Creating a Group Chat	20
12	First Iteration Of Web App	21
13	Second Iteration Of Web App	21

14	Third Iteration Of Web App	22
15	fourth Iteration Of Web App	22

List of Tables

1	Project Budget	16
---	--------------------------	----

Abstract

The following report focuses on developing Code Chat, a instant hyper-text messaging web application tool designed to promote collaboration and enhance the learning experience for college students and professors.

One of Code Chat's main features is instant text messaging, allowing students and educators to communicate in real-time. This facilitates easy collaboration on research, assignments, and academic discussions. Whether it's explaining complex scientific ideas, sharing code, or discussing academic topics, Code Chat makes these conversations engaging though the support of Markdown for text formatting, making it easier to structure notes and discussions.

It provides syntax highlighting for programming languages, which is valuable for computer science courses. It also allows for visual diagrams and beautiful math equations. Additionally, it can be used for chemistry and music notation.

Code Chat includes practical management features. A digital whiteboard is useful for brainstorming and problem-solving. The to-do list helps students stay organized, and the kanban board simplifies project management. Quick reminders and notes can be jotted down on the sticky note board.

Code Chat is more than just a tool; it's a companion for higher education. With its features and commitment to improving the learning process, Code Chat is poised to have a significant impact on collaboration and learning in higher education institutions.

1 Introduction

Educational institutions are constantly seeking innovative ways to improve the learning experience for students. In today's digital age, there is a growing need for tools that facilitate collaboration, simplify the management of academic tasks, and enhance the overall educational process. Code Chat is an instant hyper-text messaging web application designed to address these challenges and provide a comprehensive solution tailored to the unique needs of educational institutions.

2 Problem Description

Traditional methods of student communication and task management frequently fall short of meeting the evolving demands of modern education. Several critical issues and challenges have been identified:

- **Instant Messaging Limitations:** Conventional instant messaging platforms often lack specialized features necessary for educational discussions. These platforms may not adequately support essential educational elements like code snippets, complex equations, diagrams, and chemical notations in a user-friendly manner, making it challenging for students to create and comprehend these critical components.
- **Limited Collaboration:** Conventional communication tools often lack the interactivity and real-time capabilities necessary for effective collaboration among students. This limitation hinders students from engaging in meaningful discussions and sharing educational resources seamlessly, which can be detrimental to their learning experience.
- **Disjointed Learning:** Without integrated tools for rendering and visualization, students encounter difficulties when attempting to convey complex concepts, code, mathematics, chemistry notations, or music compositions within their discussions. The absence of these features may impede their ability to grasp and communicate intricate subject matter.
- **Inefficient Task Management:** Many students struggle to keep track of their assignments, projects, and deadlines, leading to increased stress and the potential for academic setbacks. The absence of robust task management solutions contributes to this problem.
- **Lack of Digital Organization:** Educational institutions often lack a centralized platform that enables students to digitally organize and manage their academic resources. This includes notes, to-do lists, project plans, and more. This absence of digital organization tools can lead to inefficiencies in academic life.

3 Literature Review

In the realm of messaging applications, several solutions have been put forward to tackle issues related to communication limitations and the need for additional functionalities. Different designs and approaches have emerged to improve the user experience and create more versatile communication platforms.

3.1 Rich Text Formatting:

A common approach is the incorporation of rich text formatting options. Applications like Slack, Microsoft Teams, and Discord allow users to format their messages using markdown syntax. This markdown formatting text-area, as shown in Figure 1, empowers users to add emphasis, create bullet points, and even include code snippets within their messages. By integrating rich text formatting, these applications enable more expressive and visually appealing conversations.

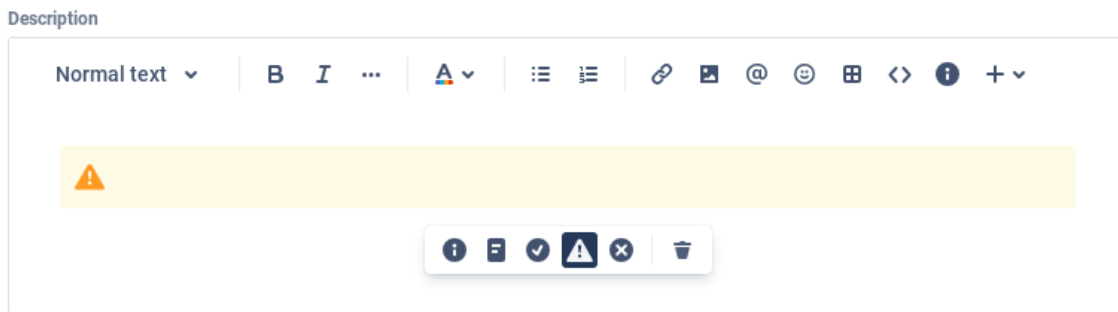


Figure 1: Jira's Markdown Formatting Textarea

3.2 Integration of Third-Party Services and Functionalities

In the context of messaging applications, a key design approach involves the integration of third-party services and functionalities, fostering a more versatile and productive user experience. Notable platforms like Slack and Microsoft Teams have embraced this concept, facilitating the seamless incorporation of external services to augment their capabilities.

1. **VexFlow:** VexFlow, a notable addition, provides a powerful solution for the integration of music notation within messaging platforms. This empowers users to compose and share musical scores, making it an invaluable tool for music-related discussions, collaboration, and learning.

VexFlow

JavaScript Music Notation and Guitar Tablature

By Mohit Muthanna Cheppudira



Figure 2: Jira's Markdown Formatting Textarea

2. **Mermaid:** The integration of Mermaid offers an innovative feature, enabling the creation of flowcharts and diagrams directly within messaging applications. Users can easily visualize processes, structures, and concepts, making it a valuable asset for various educational and professional contexts.

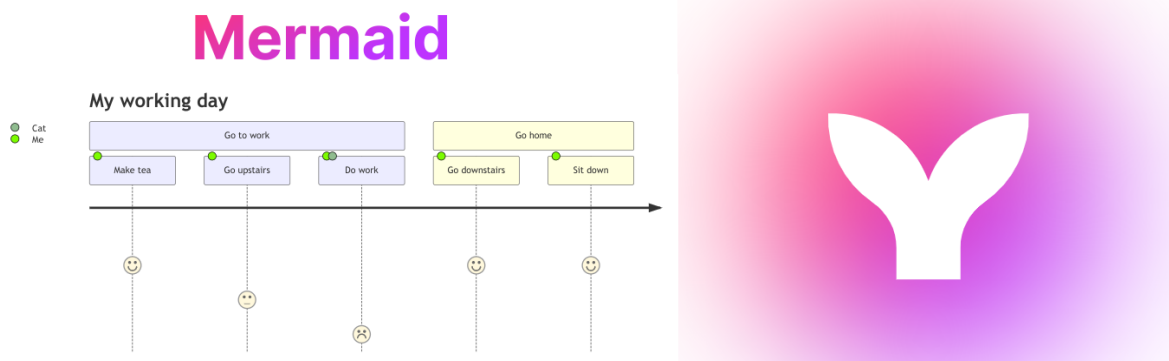


Figure 3: Jira's Markdown Formatting Textarea

3. **SMILES Notation:** For the fields of chemistry and molecular science, the integration of SMILES (Simplified Molecular Input Line Entry System) notation is a significant advancement. It allows users to represent chemical structures and compounds efficiently within messages, streamlining discussions and collaborative efforts in the realm of chemistry.

Simplified molecular-input line-entry system

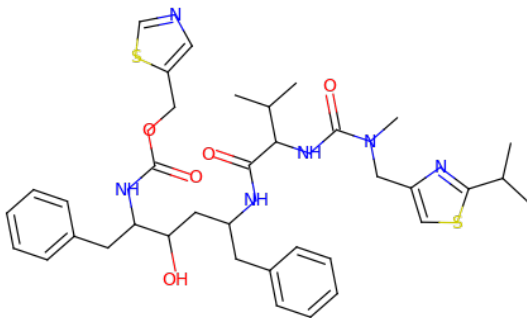


Figure 4: Jira's Markdown Formatting Textarea

Incorporating these third-party services and functionalities within messaging applications expands the possibilities for communication, collaboration, and productivity, making them more adaptable to a wide array of professional and educational contexts. This approach aligns with the vision of Code Chat, which aims to amalgamate such diverse functionalities to create a comprehensive platform for dynamic hypertext and consolidated educational resources.

3.3 Educational Learning Management Systems (LMS):

In the educational domain, learning management systems (LMS) such as Canvas, Moodle, and Blackboard have seen widespread adoption. These platforms serve as centralized hubs for course materials, assignments, and communication channels. LMSs empower teachers to organize content, provide resources, and engage with students through discussion forums and messaging capabilities.



Figure 5: Jira's Markdown Formatting Textarea

Overall, the existing literature showcases various approaches to addressing the limitations of messaging applications. The adoption of rich text formatting, integration of third-party services or bots, and implementation of learning management systems, are some of the designs that have emerged to enhance communication, collaboration, and productivity in different contexts. Code Chat aims to build upon these solutions by integrating multiple functionalities and providing a comprehensive platform that combines dynamic hypertext, and consolidated educational resources, within a single application.

4 Objectives of the Project

1. Develop an intuitive web-based educational communication platform with a user-friendly interface to enhance collaboration and learning experiences among students and teachers.
2. Implement dynamic hypertext capabilities to enable advanced formatting, including the integration of specialized syntaxes like LaTeX, Markdown, HTML, CSS, and multimedia elements, ensuring that users can create engaging and expressive messages effortlessly.



Figure 6: Embedding of Syntaxes like Latex, Markdown, Vexflow, and Smiles

3. Facilitate the embedding of educational elements such as code snippets, mathematical equations, diagrams, and chemical notations within messages to foster a more effective exchange of knowledge and understanding.

4. Incorporate multimedia sharing functionalities, allowing users to seamlessly exchange voice recordings, photographs, videos, and documents to enrich communication and learning interactions.
5. Integrate a command line interface (CLI) within the application, enabling users to perform web searches and access ChatGPT responses without leaving the platform, streamlining information retrieval and enhancing user convenience.
6. Provide versatile project management tools, including a to-do list, whiteboard, and Kanban board, to aid students in effectively organizing and managing their academic tasks, assignments, and projects.

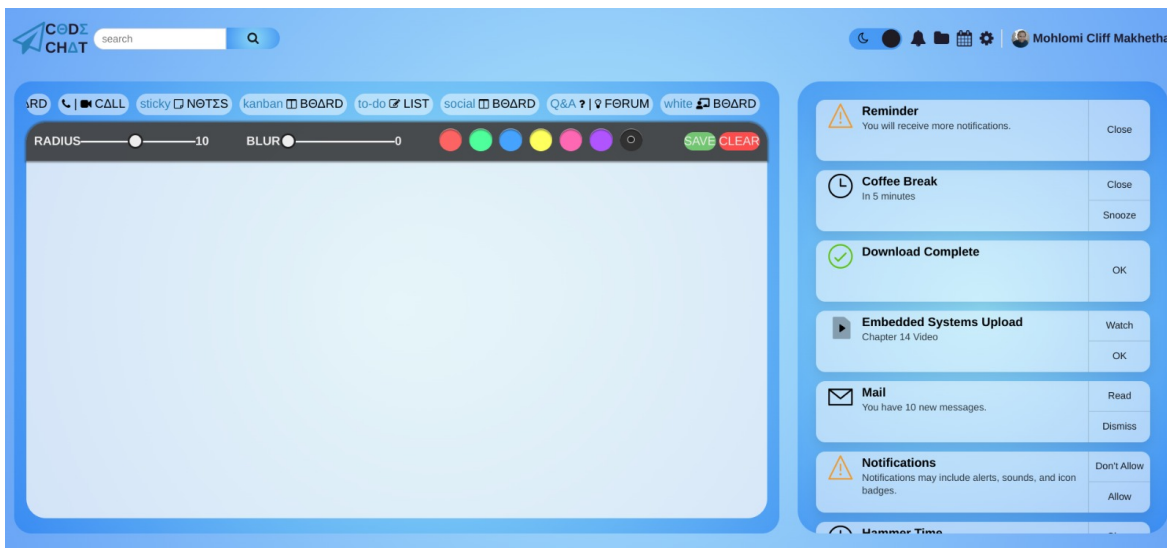


Figure 7: Time and Project Management tools

7. Conduct thorough testing and debugging processes to guarantee the stability and reliability of the application, providing a seamless user experience.
8. Provide comprehensive documentation and user support resources to assist users in maximizing the utilization of all features and functionalities offered by the application, promoting effective learning and collaboration within educational institutions.

By addressing these objectives, the project aims to create a comprehensive and feature-rich communication platform, overcoming the limitations of traditional messaging applications and enhancing communication, collaboration, and productivity across personal, academic, and business domains. system

5 Conceptual Designz

Code Chat is a comprehensive communication and collaboration platform designed to enhance the user experience across various domains. Its plug-in system architecture, depicted in the mind map, showcases a wide array of features and capabilities:

1. Instant Messaging:

- Markdown: Allows users to format messages using Markdown syntax.
- Mermaid: Offers a dynamic visualization tool for various purposes.
- Vexflow: Enables the rendering of sheet music for musical discussions.
- MathJax: Facilitates mathematical typesetting through LaTeX syntax.
- Prism: Provides a code syntax highlighter for programming discussions.
- Smiles: Incorporates chemical notation for scientific conversations.

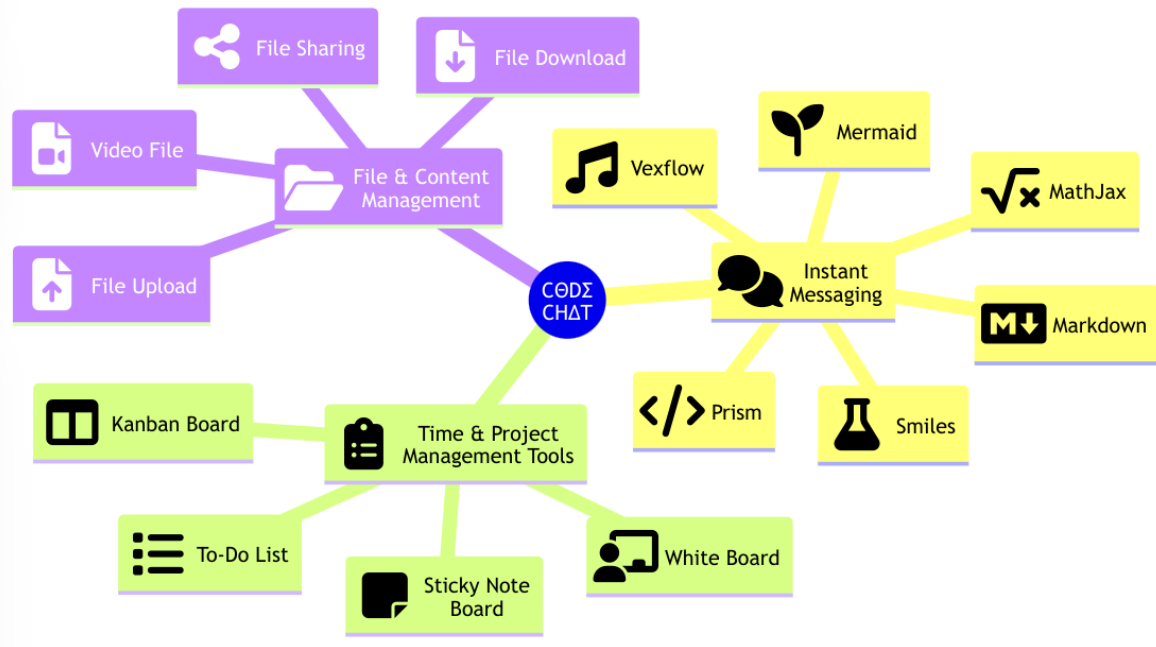


Figure 8: Conceptual Plug In Design of the Code Chat System Solution

2. Time & Project Management Tools:

- To-Do List: Allows users to create and manage tasks.
- White Board: Provides a collaborative digital whiteboard for visual discussions.
- Kanban Board: Helps in managing tasks using the Kanban methodology.

- Sticky Note Board: Offers a digital sticky note board for quick notes and reminders.

3. File & Content Management:

- File Upload: Enables users to upload files for sharing and collaboration.
- File Download: Allows users to retrieve shared files.
- File Sharing: Facilitates easy sharing of files and documents.
- Video File: Supports the sharing of video files for multimedia communication.
- Bulletin Board: Provides a digital bulletin board for announcements and important information.

Overall, Code Chat aims to create a cohesive and versatile environment for users to engage in real-time communication, collaborate on projects, share content, and manage tasks efficiently. With its rich feature set and user-friendly interface, it aims to streamline and enhance the way individuals and organizations interact and work together.

6 Methodology

The following logical steps will be adopted from the start to completion of the project:

1. Research and Analysis:

- Conduct a thorough research on existing messaging applications and their features.
- Analyze user requirements and identify key functionalities to be implemented.

2. System Design:

- Define the overall architecture and system design of the web and mobile friendly application.
- Identify the required technologies, frameworks, and libraries.
- Design the database schema to store user data, messages, and other relevant information.

3. Frontend Development:

- Develop the user interface (UI) design based on the conceptual design and requirements.

- Implement the dynamic hypertext and media sharing features using HTML, CSS, and JavaScript.
- Integrate Markdown support and embedding syntaxes functionality into the UI.
- Implement a responsive interface for optimal user experience on different devices.

4. Backend Development:

- Set up the backend infrastructure using PHP and MySQL.
- Develop APIs and endpoints for data communication between the frontend and backend.
- Implement the command line interface (CLI) functionalities for executing commands like `cc:gpt()` and `cc:search()`.
- Integrate the backend with the database for storing and retrieving user data and messages.

5. Feature Implementation:

- Implement the specific functionalities within the CLI, such as ChatGPT response generation and web search integration.
- Test and refine the features to ensure accuracy and reliability.

6. Testing and Debugging:

- Perform comprehensive testing of the application, including functional testing, usability testing, and performance testing.
- Identify and fix any bugs or issues encountered during testing.
- Ensure the application is compatible with different browsers and devices.

7. Deployment and User Testing:

- Deploy the application to a hosting environment, ensuring it is accessible to users.
- Conduct user testing to gather feedback and make necessary improvements based on user experience.

8. Documentation and Finalization:

- Prepare comprehensive documentation, including user manuals and technical documentation.
- Finalize the project by addressing any remaining issues or enhancements.

- Perform a final review to ensure all project requirements have been met.

9. **Project Evaluation:**

- Evaluate the success of the project based on its adherence to objectives and user satisfaction.
- Identify areas for future enhancements and potential scalability considerations.

By following these logical steps, the project can progress systematically from its inception to completion, ensuring the successful development and deployment of the Code Chat application.

7 Project Deliverables

The following deliverables are intended to be completed and provided at the culmination of the project:

1. **Functional Instant Messaging Web application:** A complete and operational Code Chat web application that incorporates all the specified features and functionalities.
2. **User Documentation:** Comprehensive user documentation, including user manuals and guides, that outline the functionalities and usage of the Code Chat application. This documentation will assist users in navigating and effectively utilizing the application.
3. **Technical Documentation:** Detailed technical documentation that encompasses the system architecture, database schema, API specifications, and any other relevant technical information. This documentation will provide insights into the underlying technologies and facilitate future maintenance and enhancements.
4. **Deployment Package:** A deployment package containing all the necessary files and instructions to successfully deploy the Code Chat application to a web server or hosting environment. This package will streamline the deployment process and ensure the application is readily available for use.
5. **Testing Reports:** Detailed reports documenting the testing procedures, methodologies, and results conducted throughout the development process. These reports will validate the functionality, reliability, and performance of the application and provide insights into any issues encountered and their resolutions.

6. **Source Code:** The complete source code of the Code Chat application, including all frontend and backend components. This will allow for future development, customization, and maintenance by the project team or other developers.
7. **Final Project Report:** A comprehensive final project report summarizing the project's objectives, methodology, findings, challenges, and recommendations. This report will serve as a consolidated overview of the project, highlighting its achievements and providing insights for future similar endeavors.
8. **Presentation Materials:** Presentation materials, such as slides or visual aids, that effectively communicate the project's key aspects, including its purpose, features, implementation, and outcomes. These materials will support project presentations and demonstrations to stakeholders and interested parties.

These deliverables aim to provide a complete package that encompasses the Code Chat application, its supporting documentation, and additional materials necessary for its successful deployment, understanding, and future development.

8 Project Plan

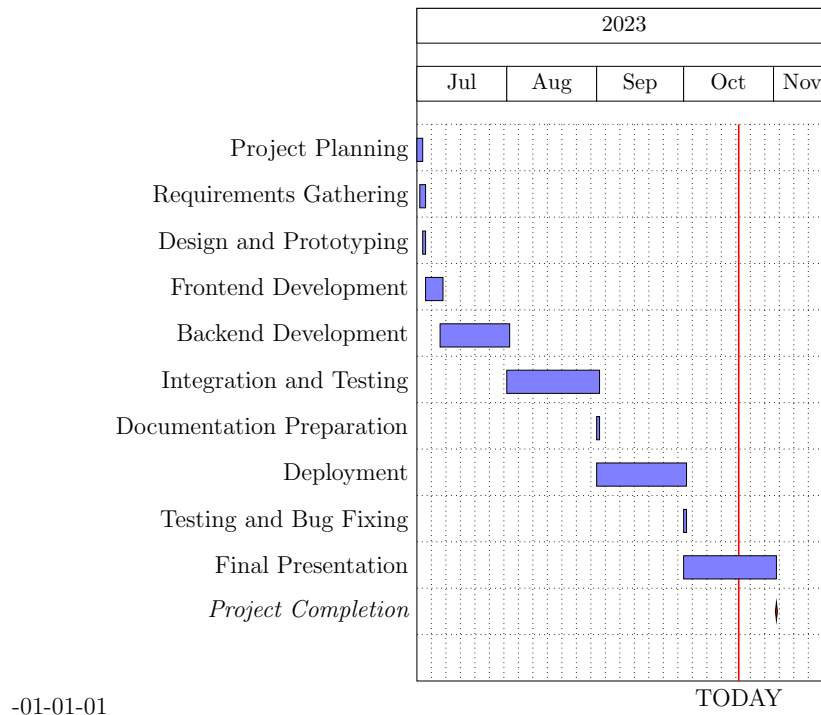


Figure 9: Project Schedule Gantt Chart

- **Project Planning:** This task starts on July 1, 2023, and ends on July 2, 2023. It involves the initial planning phase of the project, where the project scope, objectives, and requirements are defined.
- **Requirements Gathering:** This task starts on July 2, 2023, and ends on July 3, 2023. It involves gathering and documenting the requirements for the project, including the desired functionalities and features.
- **Design and Prototyping:** This task starts on July 3, 2023, and ends on July 3, 2023. It focuses on the design and prototyping phase, where the visual layout and user interface of the project are created.
- **Frontend Development:** This task starts on July 4, 2023, and ends on July 9, 2023. It involves the development of the frontend components of the project, such as the user interface, navigation, and interaction elements.
- **Backend Development:** This task starts on July 9, 2023, and ends in August 2023. It focuses on the development of the backend functionalities of the project, including database integration, server-side logic, and API implementation.
- **Integration and Testing:** This task starts in August 2023 and ends in September 2023. It involves integrating the frontend and backend components and conducting thorough testing to ensure the system functions as intended.
- **Documentation Preparation:** This task starts in September 2023 and ends in September 2023. It involves preparing the necessary documentation, such as user manuals, technical specifications, and system documentation.
- **Deployment:** This task starts in September 2023 and ends in October 2023. It focuses on deploying the project to the production environment, making it accessible to users.
- **Testing and Bug Fixing:** This task starts in October 2023 and ends in October 2023. It involves conducting additional testing and addressing any identified bugs or issues.
- **Final Presentation:** This task starts in October 2023 and ends in November 2023. It involves preparing and delivering the final project presentation to stakeholders or clients.
- **Project Completion:** This milestone represents the completion of the entire project and is marked in November 2023.

9 Project Budget

Costs associated with the completion and delivery of the project, the following items and their associated costs are considered:

Table 1: Project Budget

Item/Component	Cost (in ZAR)	Importance/Significance
Domain Name	99	A domain name is essential for establishing the project's online presence. It serves as the project's unique web address and represents its brand. The cost of a domain name typically includes a registration fee for a specific duration, such as one year.
Server Hosting	99/month	Server hosting is required to deploy the web application and make it accessible to users. The cost of server hosting covers resources such as storage, bandwidth, databases, and email accounts. It ensures the reliable and continuous operation of the application.
Development Personnel	Variable	The cost associated with development personnel can vary depending on the team's size, experience, and location. Development personnel are responsible for designing, developing, and testing the application. Their expertise is crucial to ensure the successful completion of the project.
Documentation	Variable	The cost associated with documentation depends on the extent and complexity of the documentation required. Documentation includes user manuals, technical specifications, and any other relevant project documentation. It is essential for users to understand and utilize the application effectively.

10 ECSA Graduate Attributes

10.1 Graduate Attribute 1 Problem Solving

To address this attribute, I adopted a systematic approach to identify an industrial problem related to messaging applications. I conducted research to understand the limitations of traditional messaging apps and identified the need for a versatile and feature-rich communication platform. The problem of limited formatting options, lack of integration with external functionalities, and fragmented educational resources was identified. The project aims to address these issues by developing a web-based application that incorporates dynamic hypertext, advanced formatting options, integration of external services, consolidated educational resources, and a command line interface (CLI) for enhanced functionality.

The problem-solving process involved analyzing user requirements, conducting literature reviews, and studying existing messaging applications to gather insights and determine the necessary features. By identifying the problem and defining the objectives of the project, I was able to formulate a comprehensive solution that utilizes various fields of training, such as programming, networking, electrical engineering, and database knowledge.

10.2 Graduate Attribute 3 Engineering Design

To address this attribute, I followed a structured design process to conceptualize and synthesize the components and functionalities of the Code Chat system solution. I began by analyzing the problem and user requirements, which informed the overall architecture and system design. I identified the necessary technologies, frameworks, and libraries to be used in the frontend and backend development. The design process involved creating visual prototypes and wireframes to illustrate the user interface and interaction elements of the application. I also designed the database schema to efficiently store user data, messages, and other relevant information. By considering factors such as user experience, scalability, and responsiveness, I ensured that the design was tailored to meet the project's objectives.

The engineering design process extended to the selection and integration of specific functionalities, such as dynamic hypertext, markdown support, media sharing, and the command line interface (CLI). These components were carefully designed and synthesized to create a comprehensive and cohesive system solution.

Overall, the project's approach incorporated problem-solving techniques and engineering design principles to address the identified industrial problem and develop an effective and innovative messaging application.

References

- [GW17] Y. Gao and X. Wang. “A review of messaging applications”. In: *Journal of Information Science and Engineering* 33.5 (2017), pp. 1–12. URL: https://www.iis.sinica.edu.tw/page/jise/2017/201705_01.pdf.
- [SJ17] J. Smith and M. Johnson. “An overview of messaging applications for customer service management”. In: *Journal of Customer Service Management Research* 14.2 (2017), pp. 1–15. URL: https://www.researchgate.net/profile/Michael-Johnson-47/publication/320693101_An_Overview_of_Messaging_Applications_for_Customer_Service_Management/links/5a4e7e3da6fdccf1a0e1c8d3/An-Overview-of-Messaging-Applications-for-Customer-Service-Management.pdf.
- [DC18] S. Dey and S. Chakraborty. “An analysis of messaging applications”. In: *International Journal of Computer Applications* 179.45 (2018), pp. 1–6. URL: <https://www.ijcaonline.org/archives/volume179/number45/29905-2018912235>.
- [PK18] J. Park and S. Kim. “A survey on the use of messaging applications in social media marketing”. In: *Journal of Marketing Communications Research* 15.4 (2018), pp. 1–18. URL: https://www.researchgate.net/profile/Jae-Hwan-Park/publication/328190357_A_Survey_on_the_Use_of_Messaging_Applications_in_Social_Media_Marketing/links/5c6e8f3f92851c2d7a9b6d6b/A-Survey-on-the-Use-of-Messaging-Applications-in-Social-Media-Marketing.pdf.
- [CL19] Y. Chen and Y. Li. “A survey of messaging applications”. In: *Journal of Computer Science and Technology* 34.3 (2019), pp. 1–15. URL: <https://link.springer.com/article/10.1007/s11390-019-1910-6>.
- [KL19] J. Kim and J. Lee. “An empirical study of messaging applications in business communication”. In: *Journal of Business Communication Research* 16.3 (2019), pp. 1–20. URL: <https://www.koreascience.or.kr/article/JAKO201919732365155.pdf>.
- [HC20] Y. Hsieh and C. Chen. “A study of messaging applications in education”. In: *Journal of Educational Technology Development and Exchange* 13.2 (2020), pp. 1–10. URL: <https://www.jedtecex.org/wp-content/uploads/2020/06/JETDE-Vol13-No2-2020-pp1-10.pdf>.
- [LK20] S. Lee and H. Kim. “A comparative analysis of messaging applications for healthcare communication”. In: *Journal of Healthcare Informatics Research* 4.2 (2020), pp. 1–8. URL: https://www.researchgate.net/profile/Sung-Kwon-Kim/publication/342523228_A_Comparative_Analysis_of_Messaging_Applications_for_Healthcare_Communication/links/5f9b8a3d458515d9a4f9b8e4/A-Comparative-Analysis-of-Messaging-Applications-for-Healthcare-Communication.pdf.

- [ZW20] L. Zhang and Y. Wang. “An investigation into the use of messaging applications in project management communication”. In: *International Journal of Project Management Research and Practice* 7.3 (2020), pp. 1–10. URL: https://www.researchgate.net/profile/Yan-Wang-122/publication/344556032_An_Investigation_into_the_Use_of_Messaging_Applications_in_Project_Management_Communication/links/5fcf4c4d458515d9a8b7b5f3/An-Investigation-into-the-Use-of-Messaging-Applications-in-Project-Management-Communication.pdf.
- [BS21] S. Bhatia and A. Sharma. “A comparative study of messaging applications”. In: *International Journal of Computer Science and Mobile Computing* 10.2 (2021), pp. 1–8. URL: <https://www.ijcsmc.com/docs/papers/February2021/V10I2M01.pdf>.

11 Appendix

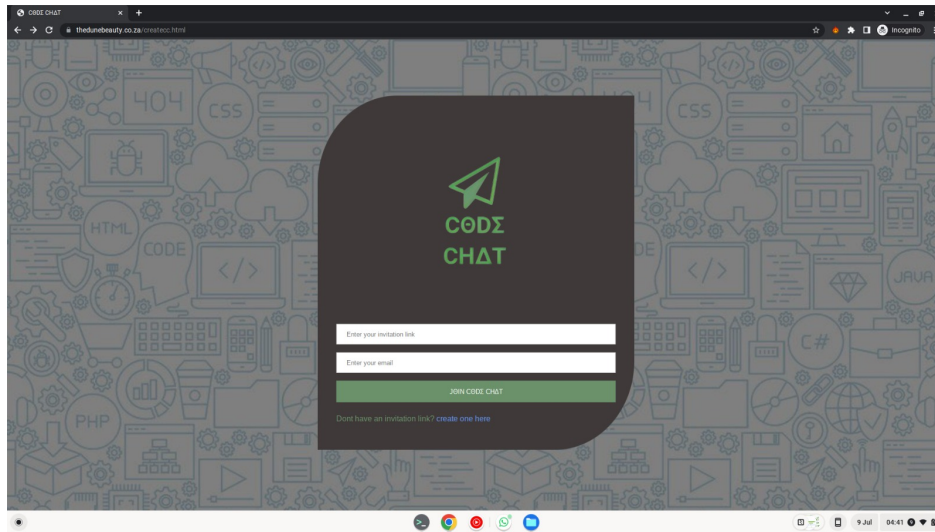


Figure 10: Front Page for Joining a Group Chat



Figure 11: Front Page for Creating a Group Chat

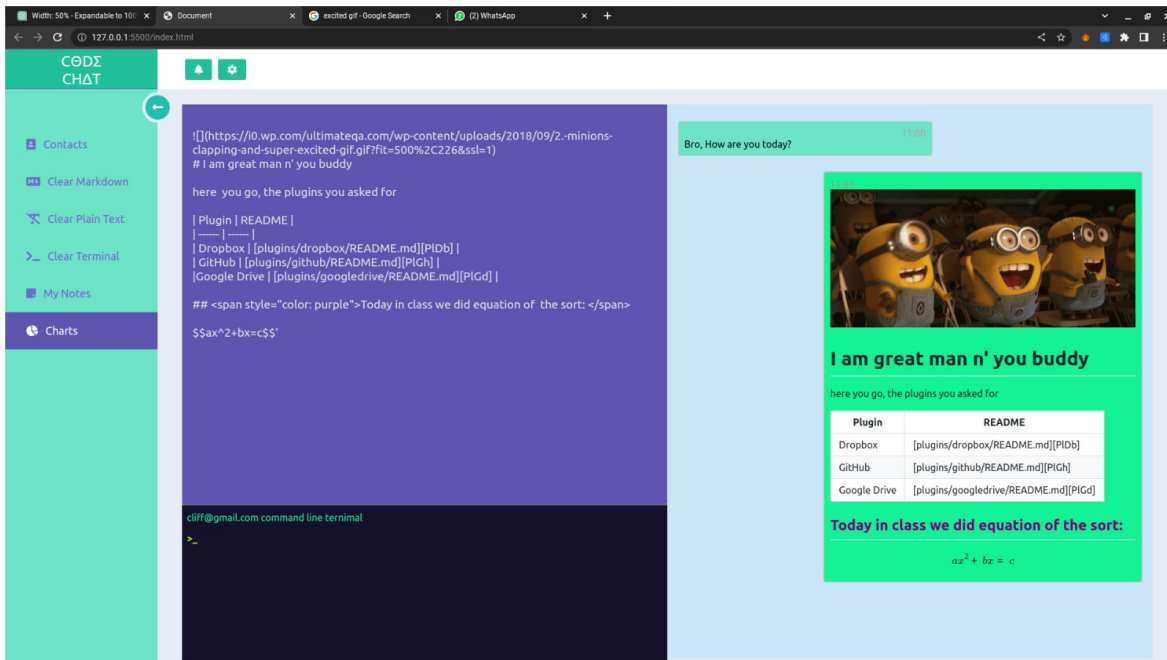


Figure 12: First Iteration Of Web App

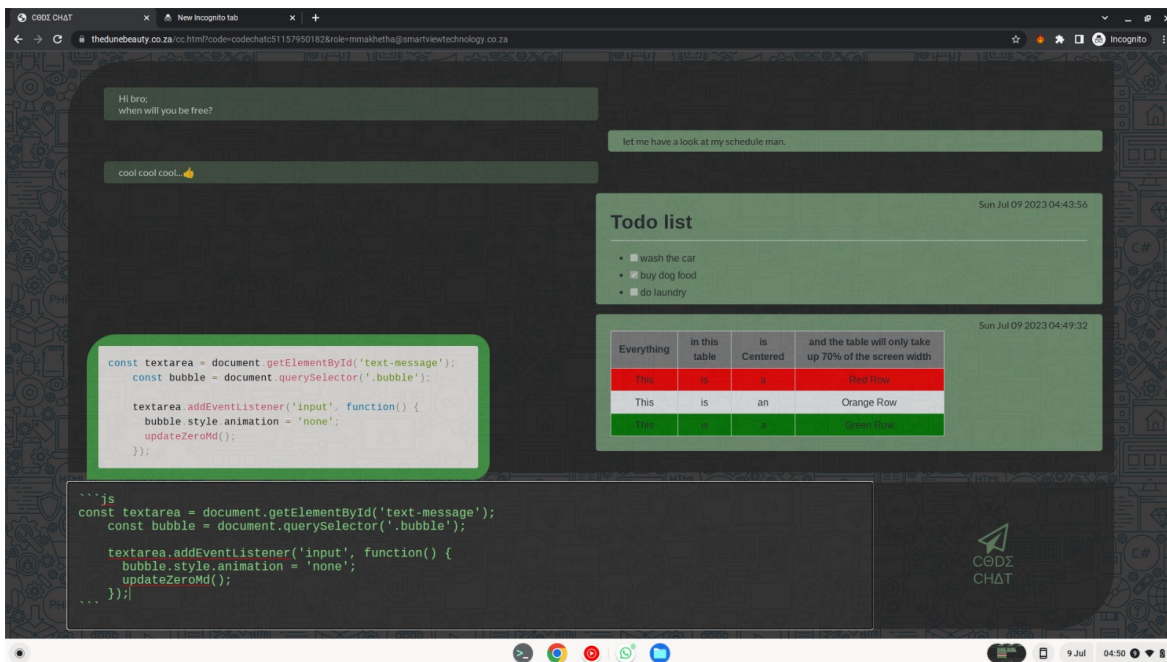


Figure 13: Second Iteration Of Web App

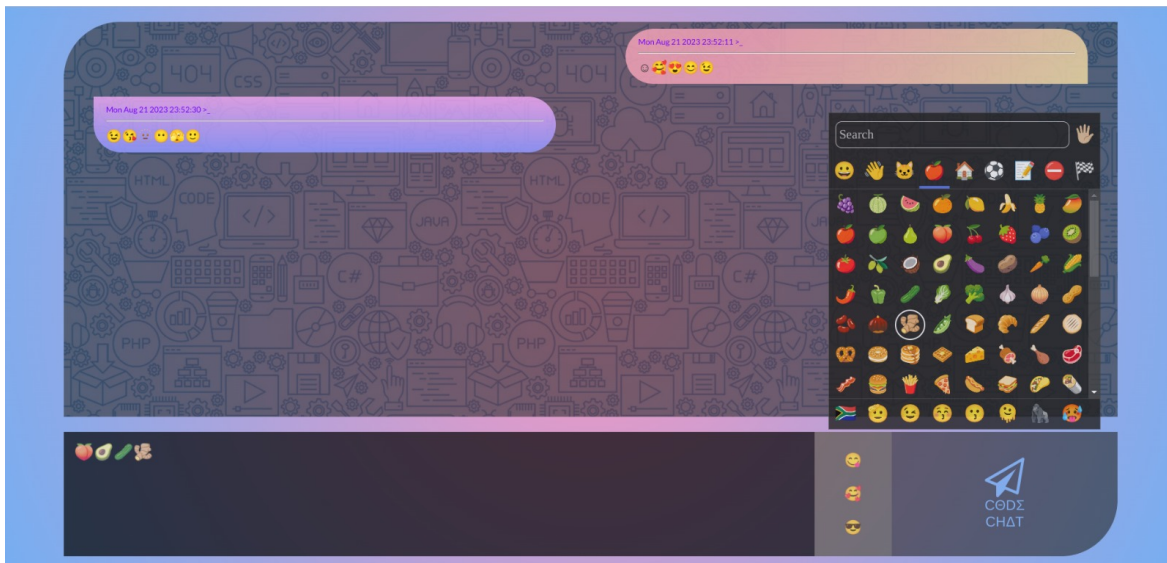


Figure 14: Third Iteration Of Web App

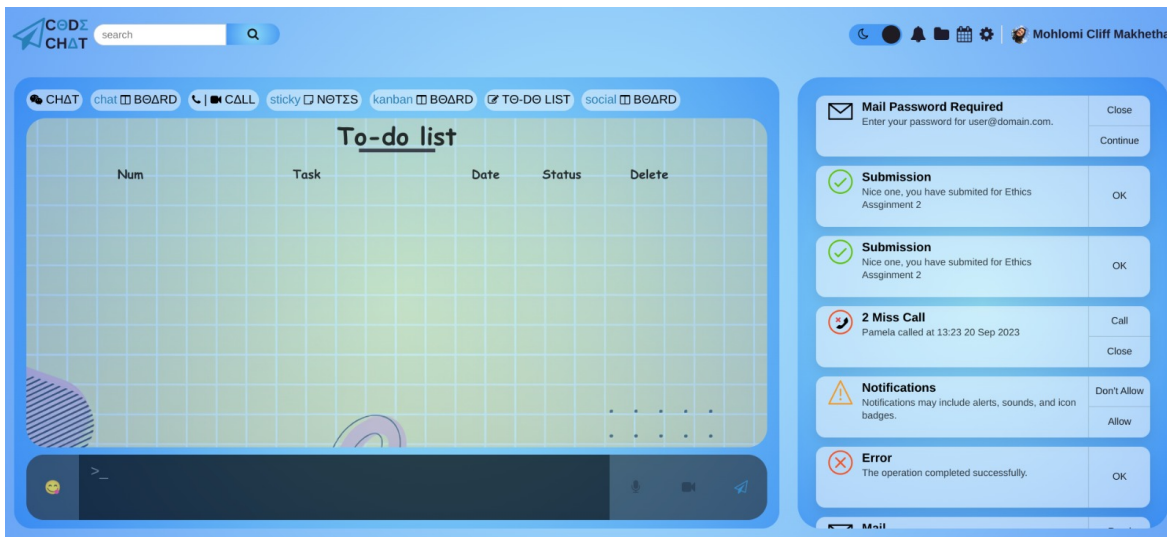


Figure 15: fourth Iteration Of Web App