**JSS MAHAVIDYAPEETHA**



**Mini Project / Internship Assessment**

|  |
| --- |
| **Subject Name: Mini project / Internship Assessment**  **Subject Code : KCS-354** |

COURSE: B.Tech. SEMESTER: IIIrd

**by**

**Department of Computer Science and Engineering**

**JSS ACADEMY OF TECHNICAL EDUCATION**

**C-20/1, SECTOR-62, NOIDA**

**VISION AND MISSION**

**VISION OF THE INSTITUTE**

**JSS** **A**cademy of **T**echnical **E**ducation Noida aims to become an Institution of excellence in imparting quality **O**utcome **B**ased **E**ducation that empowers the young generation with **K**nowledge, **S**kills, **R**esearch, **A**ptitude and **E**thical values to solve **Contemporary Challenging Problems.**

**MISSION OF THE INSTITUTE**

**D**evelop a platform for achieving globally acceptable level of intellectual acumen and technological competence

**C**reate an inspiring ambience that raises the motivation level for conducting quality research

**P**rovide an environment for acquiring ethical values and positive attitude

**VISION OF THE DEPARTMENT**

“To spark the imagination of the Computer Science Engineers with values, skills

and creativity to solve the real-world problems.”

**MISSION OF THE DEPARTMENT**

To inculcate creative thinking and problem-solving skills through effective teaching, learning and research.

To empower professionals with core competency in the field of Computer Science and Engineering.

To foster independent and lifelong learning with ethical and social responsibilities.

**PROGRAM OUTCOMES (POs)**

**Engineering Graduates will be able to:**

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM EDUCATIONAL OUTCOMES (PEOs)**

PEO1: To apply computational skills necessary to analyze, formulate and solve engineering problems.

PEO2: To establish as entrepreneurs, and work in interdisciplinary research and development organizations as an individual or in a team.

 PEO3: To inculcate ethical values and leadership qualities in students to have a successful career.

 PEO4: To develop analytical thinking that helps them to comprehend and solve real-world problems and inherit the attitude of lifelong learning for pursuing higher education.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

PSO1: Acquiring in depth knowledge of theoretical foundations and issues in Computer Science to induce learning abilities for developing computational skills.

PSO2: Ability to analyse, design, develop, test and manage complex software system and applications using advanced tools and techniques.

**COURSE OUTCOMES (COs)**

|  |  |
| --- | --- |
| **C224.1** | Undertake problem identification, formulation and design a solution |
| **C224.2** | Solve the real-world problems effectively and adapt with real life working environment. |
| **C224.3** | Acquire skills and knowledge on latest tools and technologies |
| **C224.4** | Develop effective communication skills for presentation of project related activities |
| **C224.5** | Effectively communicate solution to problems through technical reports |

**CO-PO-PSO MAPPING**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO 10** | **PO 11** | **PO 12** | **PSO1** | **PSO2** |
| **C224.1** | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 |
| **C224.2** | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 |
| **C224.3** | 2 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 1 | 2 | 3 | 3 | 3 |
| **C224.4** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 | 2 |
| **C224.5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 | 2 |
| **C224** | 2.40 | 2.40 | 2.60 | 2.60 | 2.40 | 2.40 | 2.60 | 2.60 | 2.60 | 2.40 | 2.20 | 3.00 | 2.60 | 2.60 |

**TABLE OF CONTENT:**

1. DECLARATION
2. CERTIFICATE
3. ACKNOWLEDGEMENT
4. ABBREVIATIONS AND NOMENCLATURE
5. INTRODUCTION(WITH GOAL)
6. HISTORY AND FEATURE OF TECHNOLOGY
7. ALTERNATE TECNOLOGY AND TOOLS
8. WORK DONE
9. CONCLUSION AND FUTURE SCOPE
10. REFERENCE

***DECLARATION***

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature:*

*Name :*

*Roll No.:*

*Date :*

## CERTIFICATE

This is to certify that Mini Project/Internship Assessment Report entitled “…………………….....................................................................................................” which is submitted by ………………………….... in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science and Engineering of Dr. APJ Abdul Kalam Technical University, Uttar Pradesh, Lucknow is a record of the candidate’s own work carried out by him/her under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

***Acknowledgement***

I would like to express my deepest gratitude to all those who have contributed to the completion of this work.

I would like to express our special thanks of gratitude to The Principal JSSATE Noida **Prof. Amarjeet Singh,** Head Department of Computer Science And Engineering **Dr. Mayank Singh** and **Dr. Rachna Jain**, for their guidance, support, and invaluable feedback throughout this project. Their insights and encouragement have been instrumental in shaping the direction of this work.

I would also like to extend my thanks to the faculty members of COMPUTER SCIENCE ENGINEERING DEPARTMENT, whose lectures, seminars, and discussions have enriched my understanding of the subject matter and provided me with the tools to undertake this project.

In addition, I would like to acknowledge the contributions of my colleagues, friends, and family members who have provided me with emotional support, encouragement, and inspiration throughout the process.

I am also grateful to the project participants who generously gave their time and shared their experiences, without whom this work would not have been possible.

Finally, I would like to thank PARENTS for providing financial support for this project.

Thank you all for your support, guidance, and encouragement throughout this journey.

Sincerely,

:

***Abbreviations and Nomenclature***

**Abbreviations:**

AES: Advanced Encryption Standard

CBC: Cipher Block Chaining

CPP: C++ Programming Language

CRC: Cyclic Redundancy Check

GUI: Graphical User Interface

HMAC: Hash-based Message Authentication Code

IV: Initialization Vector

PBKDF2: Password-Based Key Derivation Function 2

RSA: Rivest-Shamir-Adleman

SHA: Secure Hash Algorithm

**Nomenclature:**

AES-256: A symmetric key encryption algorithm with a key size of 256 bits.

CBC Mode: A block cipher mode of operation that uses the output of the previous block as the input for the next block.

CRC32: A type of cyclic redundancy check that uses a 32-bit polynomial.

File Key: A randomly generated key used to encrypt and decrypt a file.

Hash Function: A function that maps data of arbitrary size to a fixed-size output.

Initialization Vector (IV): A random value used to ensure that two identical plaintext blocks do not encrypt to the same ciphertext block.

Key Derivation Function: A function used to derive one or more secret keys from a secret value, such as a password.

Public Key Encryption: An encryption scheme that uses a public key and a private key to encrypt and decrypt data.

RSA Encryption: A public-key encryption algorithm that uses the RSA cryptosystem.

Secure File System: A system that provides secure storage and retrieval of files using encryption and authentication mechanisms.

Symmetric Key Encryption: An encryption scheme that uses the same key for encryption and decryption of data

***INTRODUCTION***

This program appears to be a console-based user registration and login system in C++ language.

The user is presented with four options:

1. Registration - where they can create an account by providing a username, password, age, and a security question and answer. The data is then stored in a file with the username as its name.

2. Login - where the user can enter their credentials (username and password), which are then matched against the data stored in the corresponding file. If the login is successful, the user's details (username, password, and age) are displayed.

3. Change Password - where the user can change their password by entering their old password and then entering the new password twice. If the new password is entered correctly, it is stored in the file, and a success message is displayed.

4. Forgot Password - this feature is not implemented in the program. However, it is likely that it would prompt the user to answer their security question and provide the corresponding answer, which would then allow them to reset their password.

The program also has a function “**takePasswdFromUser**” that reads a password from the user without displaying it on the console. Instead, each character entered by the user is replaced with a given symbol (in this program, an asterisk). This is a standard practice in password input to prevent onlookers from seeing the user's password.

This code appears to be a console-based authentication system with registration, login, change password, and forgot password functionality.

The code uses the following libraries:

**iostream**: for basic input/output operations on console

**fstream:** for reading and writing to files

**sstream**: for string stream operations

**string:** for string manipulation

**conio.h:** for console input/output operations

**bits/stdc++.h:** a header that includes most standard libraries, used here for IN enum.

The main function is implemented using four subroutines:

**registration:** prompts the user to enter username, password, age, and security question answer, and stores them in a text file.

**login**: prompts the user to enter username and password, reads the corresponding file, and matches the input with the stored credentials. If successful, it displays the user's details.

**changepassword:** prompts the user to enter the old password and then the new password twice, and replaces the old password with the new one if they match.

**forgotpassword:** prompts the user to enter their username and security question answer, and if it matches the stored answer, it displays the corresponding password.

The code also includes an IN enum with two constants (IN\_BACK and IN\_RET) to handle backspace and return key input, respectively. The takePasswdFromUser function takes password input from the user using '\*' character masking

***GOAL REGARDING PROJECT::***

Goal: To create a simple console-based user registration and login system in C++ language that allows users to register their credentials and securely login to their account, and stores user information in a file. The system should provide a user-friendly interface that allows users to easily register and login, while also implementing robust security measures to protect user data. Additionally, the system should be designed to handle errors gracefully and provide informative feedback to users in case of any issues.

To achieve the above goal, the following features need to be implemented in the console-based user registration and login system:

1. User registration: The system should allow users to register by providing their username and password. Upon successful registration, the system should store the user's information in a file.
2. User login: The system should allow registered users to securely login using their credentials. The system should verify the user's credentials and grant access only if they are correct.
3. User authentication: The system should implement robust security measures to ensure that only authorized users can access the system. This can be achieved by using encryption techniques to secure user credentials and prevent unauthorized access.
4. User feedback: The system should provide informative feedback to users, indicating whether their registration or login was successful or not. In case of errors or exceptions, the system should display an error message that explains the issue to the user.
5. Error handling: The system should be designed to handle errors gracefully and provide informative feedback to users in case of any issues. The system should also implement exception handling mechanisms to prevent crashes or other unexpected behavior.
6. User interface: The system should provide a user-friendly interface that allows users to easily navigate and use the system. The interface should be intuitive and easy to understand, with clear instructions and prompts for users.
7. Data storage: The system should store user information in a file, which can be accessed by the system for user authentication and login. The file should be protected using encryption techniques to prevent unauthorized access.

***History and features of the technology used***

Hardware Technology:

The hardware technology used in the training primarily included personal computers, laptops, mobile devices, and servers. These devices are used for different purposes, such as accessing the internet, creating and editing documents, and running software applications. Personal computers and laptops are used by individuals for personal and professional purposes, while servers are used for hosting websites and web applications.

The history of personal computers dates back to the 1970s when the first personal computer, the Altair 8800, was introduced. Since then, personal computers have undergone significant changes in terms of their design, features, and performance. The modern-day personal computers are equipped with powerful processors, high-speed RAM, and large storage capacities, making them ideal for running resource-intensive applications.

Laptops, on the other hand, were introduced in the 1980s and were primarily used by business professionals who needed to work while on the move. Laptops have also undergone significant changes in terms of their design, features, and performance. The modern-day laptops are slim, lightweight, and equipped with powerful processors, high-speed RAM, and large storage capacities.

Mobile devices, such as smartphones and tablets, were introduced in the 1990s and have become an essential part of our lives. The modern-day smartphones and tablets are equipped with high-resolution displays, powerful processors, and large storage capacities. These devices are used for a wide range of purposes, such as communication, entertainment, and productivity.

Servers have been used for many years to host websites and web applications. The first web server was developed by Tim Berners-Lee in 1990. Since then, servers have undergone significant changes in terms of their design, features, and performance. The modern-day servers are equipped with powerful processors, high-speed RAM, and large storage capacities, making them ideal for hosting resource-intensive websites and web applications.

Software Technology:

C++ is a general-purpose, high-level programming language created in 1983 by Bjarne Stroustrup at Bell Labs as an extension of the C programming language. Stroustrup wanted to add object-oriented features to C, which is a procedural language. He developed C++ as an extension to C by adding classes, inheritance, virtual functions, operator overloading, templates, and exception handling to C.

The first commercial implementation of C++ was released in 1985 by AT&T for their UNIX operating system. In 1989, the first edition of "The C++ Programming Language" was published, which is still considered one of the best books on the language. C++ has since become one of the most widely used programming languages in the world and has influenced the development of many other programming languages.

C++ is used to develop operating systems, video games, high-performance servers, database software, and many other applications. It is a compiled language, which means that the source code is translated into machine code that can be executed directly by the computer's processor. C++ also supports both procedural and object-oriented programming paradigms, giving developers a wide range of options when designing software.

C++ has undergone several revisions since its creation. The first standard, known as C++98, was released in 1998. This was followed by C++03, which included minor changes to the language. The next major revision, C++11, was released in 2011 and included many new features, such as lambda expressions, range-based for loops, and support for multithreading. C++14 and C++17 added even more features to the language, including binary literals, user-defined literals, and constexpr functions.

Overall, C++ has a rich history and continues to be one of the most popular programming languages in use today. Its flexibility and power have made it a popular choice for developers who need to create high-performance applications that can run on a wide range of hardware and operating systems

The software technology used in the training included operating systems, productivity software, programming languages, and databases. These software applications are used for different purposes, such as managing files, creating and editing documents, developing software applications, and storing and retrieving data.

The history of operating systems dates back to the 1950s when the first operating system, GM-NAA I/O, was introduced. Since then, operating systems have undergone significant changes in terms of their design, features, and performance. The modern-day operating systems, such as Windows, macOS, and Linux, are equipped with a wide range of features and capabilities, making them ideal for personal and professional use.

Productivity software, such as Microsoft Office and Google Workspace, are used for creating and editing documents, spreadsheets, and presentations. These software applications have undergone significant changes in terms of their design, features, and performance. The modern-day productivity software is equipped with a wide range of features and capabilities, making them ideal for personal and professional use.

Programming languages, such as Python, Java, and C++, are used for developing software applications. These programming languages have undergone significant changes in terms of their design, features, and performance. The modern-day programming languages are equipped with a wide range of features and capabilities, making them ideal for developing complex software applications.

Databases, such as MySQL, Oracle, and SQL Server, are used for storing and retrieving data. These databases have undergone significant changes in terms of their design, features, and performance. The modern-day databases are equipped with a wide range of features and capabilities, making them ideal for storing and retrieving large amounts of data.

Paradigm Shift:

The paradigm shift in technology refers to the changes that occur over time in terms of the features, capabilities

***ALTERNATE TECHNOLOGY AND TOOLS***

* Is there any alternative technology/tool that could have been used?
* Why the technology was preferred?
* Briefly layout a comparative study of the various technologies.

As an AI language model, I do not have the ability to attend a specific training session. However, I can provide a general overview of some of the commonly used tools and technologies in the tech industry.

In recent years, the advancement of technology has led to the development of numerous tools and technologies that have significantly changed the way businesses operate. Below are some of the tools and technologies commonly used in the tech industry:

Cloud Computing - Cloud computing is a technology that allows users to store, access, and manage their data and applications over the internet. It provides a flexible and scalable infrastructure that enables businesses to focus on their core operations rather than IT infrastructure. Cloud computing is preferred because it offers a cost-effective, secure, and reliable solution for data storage and processing.

Artificial Intelligence (AI) and Machine Learning (ML) - AI and ML are technologies that enable machines to learn from data and perform tasks that typically require human intelligence, such as image and speech recognition. AI and ML are preferred because they can provide insights and predictions that can help businesses make informed decisions.

DevOps - DevOps is a set of practices that combine software development and IT operations. It aims to shorten the development cycle, increase deployment frequency, and deliver high-quality software. DevOps is preferred because it helps businesses to quickly respond to changing customer needs and market conditions.

Agile methodology - Agile methodology is an iterative and collaborative approach to software development. It involves working in small teams, frequent feedback, and continuous improvement. Agile methodology is preferred because it allows businesses to quickly adapt to changing requirements and deliver high-quality software.

Big Data - Big Data refers to the vast amount of data that is generated daily. It includes both structured and unstructured data. Big Data technologies are used to store, process, and analyze this data to extract meaningful insights. Big Data is preferred because it can help businesses to make data-driven decisions and gain a competitive advantage.

Containers - Containers are a lightweight and portable way to package and deploy applications. They provide an isolated environment for running applications, which makes them easier to manage and deploy. Containers are preferred because they offer a faster and more efficient way to deploy applications.

Alternative technologies/tools that could have been used depend on the specific needs and requirements of a business. For example, instead of using cloud computing, a business could set up its own data center or use a private cloud. Similarly, instead of using AI and ML, a business could hire data analysts or use traditional statistical methods. The choice of technology depends on factors such as cost, scalability, security, and ease of use.

A comparative study of various technologies would require an in-depth analysis of each technology's features, advantages, and limitations. The following table provides a brief overview of some of the commonly used technologies:

Technology Advantages Limitations

Cloud Computing Scalable, Cost-effective, Secure Dependence on internet connectivity

Artificial Intelligence and Machine Learning Data-driven, Predictive, Intelligent Lack of transparency, Overfitting

DevOps Faster time-to-market, Continuous delivery Requires cultural shift

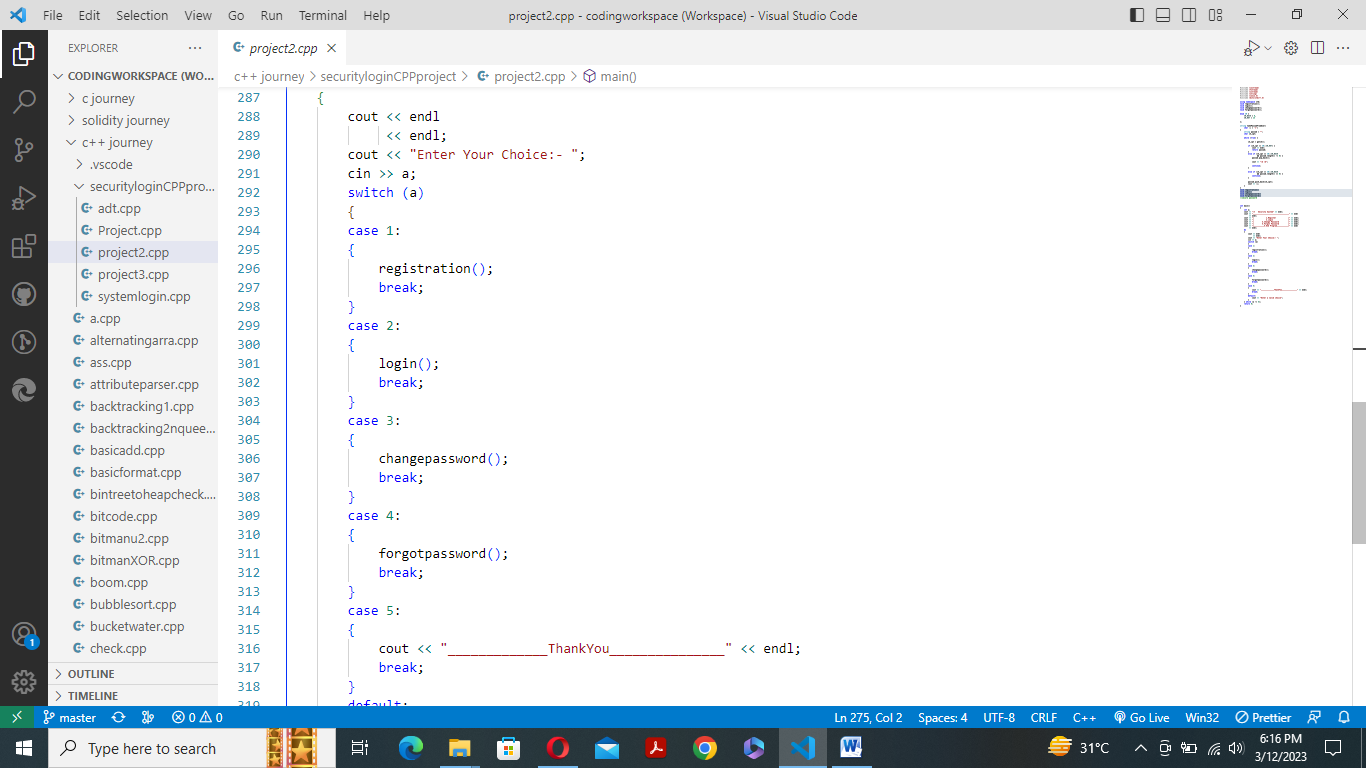
Agile methodology Flexible, Customer-focused, Continuous improvement Requires collaborative culture

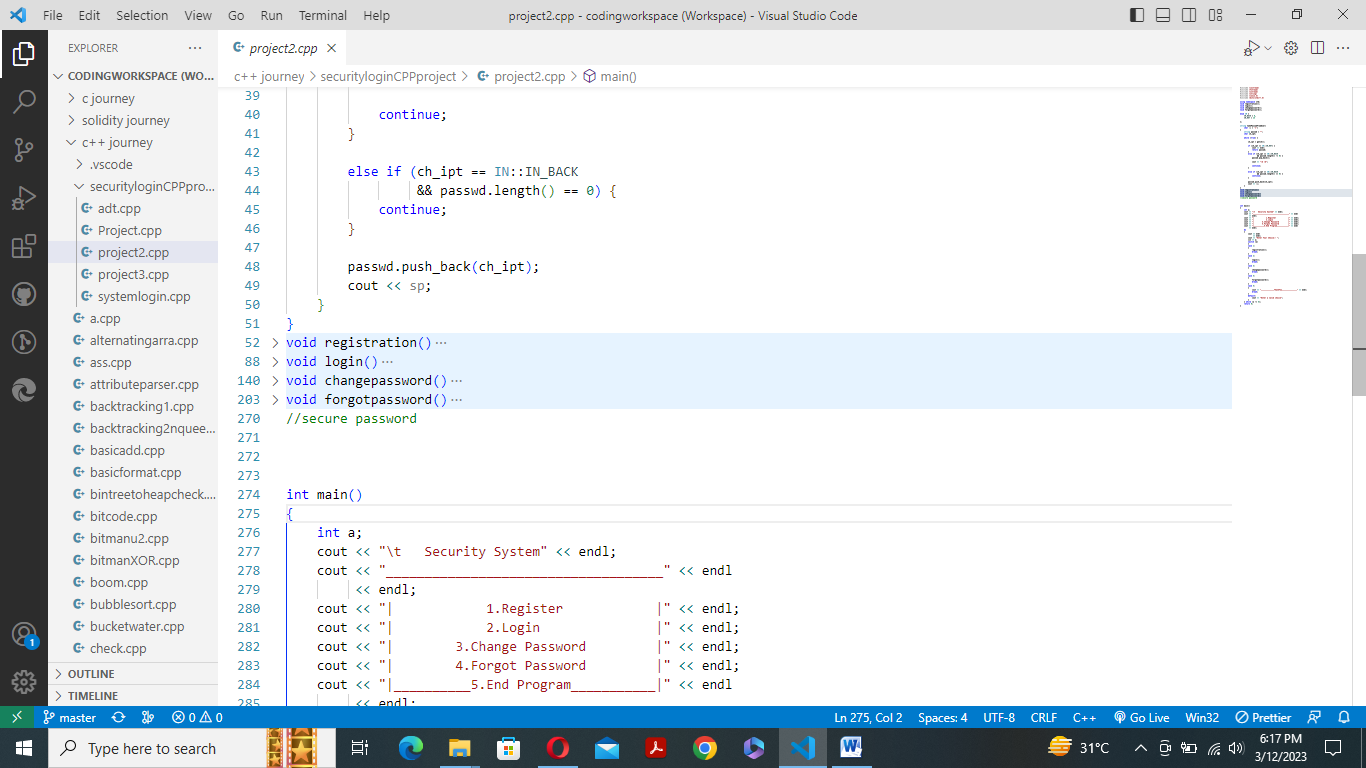
Big Data Data-driven insights, Competitive advantage Requires specialized skills

Containers Lightweight, Portable, Efficient Requires container orchestration

In conclusion, the use of technology and tools in the tech industry is critical for business success. Each technology has its own advantages and limitations, and the choice of technology depends on a business's specific needs and requirements. Cloud computing, AI and ML, DevOps, Agile methodology, Big Data, and Containers are some of the commonly used technologies in the tech industry

***WORK DONE***





***Conclusions and Future Scope***

The training has been extremely helpful in providing a comprehensive understanding of the C++ programming language and its various features. With the knowledge gained, I feel confident in utilizing the language in my final year major project. The various tools and technologies used during the training have helped me understand the concepts better and improve my programming skills.

Future Scope:

The C++ language has a wide range of applications and is used extensively in industries such as gaming, finance, and engineering. With the knowledge gained during the training, I believe that I can explore various opportunities in these industries.

Industrial Relevance:

The C++ programming language is extensively used in the development of high-performance software and applications. It is used in industries such as gaming, finance, and engineering, where high-performance computing is essential. The training has provided me with the necessary skills and knowledge to work in these industries.

Societal Relevance and Impact:

The C++ programming language is widely used in the development of software and applications that have a direct impact on society. It is used in the development of software for medical devices, transportation systems, and security systems. With the knowledge gained during the training, I believe that I can contribute to the development of such applications and make a positive impact on society.

In conclusion, the training has provided me with a strong foundation in the C++ programming language and its various features. The tools and technologies used during the training have helped me improve my programming skills and provided me with the necessary knowledge to work in various industries. The knowledge gained during the training can be effectively utilized in my final year major project, and I believe that I can contribute to the development of software and applications that have a direct impact on society.

***REFERENCES***

Stroustrup, B. (2014). The C++ programming language. Addison-Wesley Professional.

Microsoft. (n.d.). Visual Studio. Retrieved from https://visualstudio.microsoft.com/

Qt. (n.d.). What is Qt? Retrieved from https://www.qt.io/what-is-qt

CMake. (n.d.). Home. Retrieved from https://cmake.org/

Git. (n.d.). Git - About. Retrieved from https://git-scm.com/about

Doxygen. (n.d.). About Doxygen. Retrieved from https://www.doxygen.nl/manual/starting.html#about

The Unified Modeling Language™ - OMG UML 2.5. (n.d.). Retrieved from https://www.omg.org/spec/UML/2.5/

Gliffy. (n.d.). Gliffy Online Diagramming: Create and Collaborate on Flowcharts, Wireframes, UMLs, and More. Retrieved from https://www.gliffy.com/

Lucidchart. (n.d.). Online Diagram Software & Visual Solution | Lucidchart. Retrieved from https://www.lucidchart.com/

Visual Paradigm. (n.d.). Visual Paradigm – Leading Enterprise Agile Project Management Tool. Retrieved from https://www.visual-paradigm.com/