IDE(integrated development environment)

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTAR OF COMPUTER APPLICATION
(1'st Semester)

SUBMITTED BY:
Karan Kumar Mishra

SUBMITTED TO:
Ms. Nisha Chaudhary

(Assistant Professor)
Avviare Educational Hub, Noida (U.P)

AVVIARE EDUCATIONAL HUB, NOIDA
(GLOCAL UNIVERSITY, SAHARANPUR, UTTAR PRADESH)

ACKNOWLEDGEMENT

I/WE are highly grateful to the Ms. Kanika Singh, Director of Operations, Avviare Educational Hub, Noida for providing this opportunity to carry out the major project work.

The constant guidance and encouragement received from Mr. Ashutosh Rathore H.O.D. IT Department, Avviare Educational Hub, Noida has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I/WE would like to express a deep sense of gratitude and thanks profusely to Ms. Nisha Choudhary, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

I/WE express gratitude to other faculty members of IT department of Avviare Educational Hub, Noida for their intellectual support throughout the course of this work.

Finally, I/WE are indebted to all whosoever have contributed in this report work.

Names

Date

TABLE OF CONTENTS(SAMPLE)

Contents	Page No.

Note: The report of respective project should be as per prescribed format and in the same order though if some of the points are not applicable in regard with the concerned project, they might be omitted.

Format and Specification details:

Format for Major Project Report

Title page

- Abstract Acknowledgement
- List of Figures
- List of Tables
- Table of Contents

Chapter 1 Introduction

1. Introduction to Project
2. Project Category(Internet based, Application or System Development,
3. Research based ,Industry Automation, Network or System Administration)
4. Objectives
5. Problem Formulation
6. Identification/Reorganization of Need
7. Existing System
8. Proposed System
9. Unique Features of the System

Chapter 2. Requirement Analysis and System Specification

2.1 Feasibility study (Technical, Economical, Operational)
2.2 Software Requirement Specification Document which must include the following:
(Data Requirement, Functional Requirement, Performance Requirement ,Dependability Requirement, Maintainability requirement, Security requirement, Look and feel requirement)
2.3 Validation
2.4 Expected hurdles

Chapter 3. System Design

3.1 Design Approach (Function oriented or Object oriented)
3.2 Detail Design
3.3 System Design using various Structured analysis and design tools such as : DFD's,Data Dictionary,Structured charts,Flowcharts or UML
3.4 User Interface Design
3.5 Database Design
3.5.1ER Diagrams

INTRODUCTION TO PROJECT

In the realm of software development, the command-line interface (CLI)serves as a fundamental tool for interacting with computer systems and executing tasks efficiently. The project at hand, titled "IDE" is a C++ based CLI application designed to [briefly describe the main purpose or functionality of your project]. This report aims to elucidate the development,features, and significance of the project, providing an in-depth exploration of its architecture, functionality, and the underlying principles that govern its operation.

Project Category(Internet based, Application or System Development

It's great to hear that I have created an IDE for code run and compile with lots of features in C++. An IDE (Integrated Development Environment) is a software application that provides comprehensive facilities to computer programmers for software development. This IDE operates in CLI mode, offering a highly user-friendly experience. Unlike traditional command-line interfaces such as Linux or CMD, users do not need to manually input commands for specific tasks. Instead, the IDE presents a menu of options, allowing users to easily choose the desired action, such as opening a file or compiling code.

Research based ,Industry Automation, Network or System Administration)

Objectives

The objective of this project is to introduce an
IDE in CLI mode that is user-friendly.
I have incorporated numerous features into this IDE.

Features:

1. Create a new file
2. Open existing files
3. Update files
4. Delete files
5. Open files in external software such as Notepad and VSCode
6. Run and compile code written in
various languages (e.g., C, C++, Java, Python, Node.js, etc.)
7. Customize text color, background color, and other display settings

PROBLEM FORMULATION

In the development of this project, we have encountered numerous challenges. Creating a CLI-mode Integrated Development Environment (IDE) for this type of software has proven to be quite difficult. To enhance user-friendliness, we have implemented several features.For instance, when a user opens a new file or accesses an existing one,a new tab is opened for that task. Additionally, for user convenience, we have incorporated a feature to seamlessly open and edit files in other software such as Notepad and VSCode.

Unique Features of the System

1.create a new file

2.open file

3.update file

4.delete file

5.open file in other software like notepad and VScode

6.run and compile the code of any lanuage like c,c++,java,python,nodejs etc.

7.change the text color and background color etc.

Chapter 2. Requirement Analysis and System Specification

In this chapter, we delve into the critical aspects of requirement analysis and system specifications for "IDE" To ensure smooth performance and accessibility on a wide range of Windows system, the following specifications have been identified:

1. **Windows OS Compatibility:** IDE is designed to run on Windows OS, with compatibility extending from older versions to the latest. It ensures that users with varying Windows system can enjoy the IDE.

2. **Minimum Hardware Requirements:**
   RAM: A minimum of 1GB RAM is required to ensure smooth IDEplay and responsiveness.
   Internal Memory: The IDE demands approximately 127kb of space in the internal memory of the Windows system for installation and operation.

These specifications are critical to accommodate a broad user base, making IDE accessible to a wide range of Windows system owners. The chapter delves deeper into the technical aspects, ensuring that the system meets the necessary requirements for optimal performance.

CHAPTER 3
SYSTEM DESIGN

-Design Approach

In developing IDE for my MCA first sem college project, I employed an object-oriented design approach using c++.
This methodology offered several advantages:

- **Modularity and Reusability:** The object-oriented approach organized the code into manageable modules, enhancing reusability for future development.

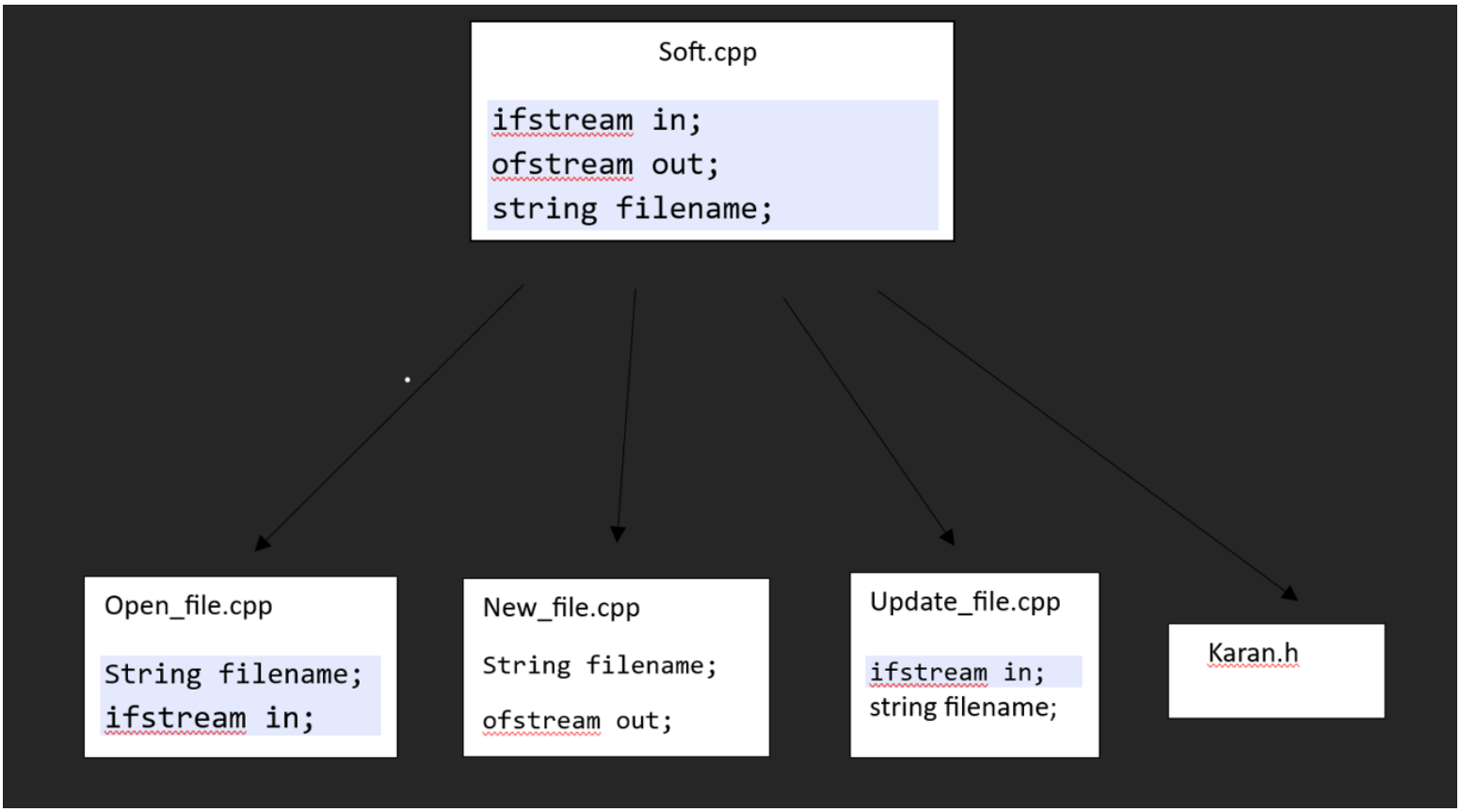
- **Encapsulation:** c++ object-oriented design encapsulated data and functions within classes, ensuring data integrity and code reliability.

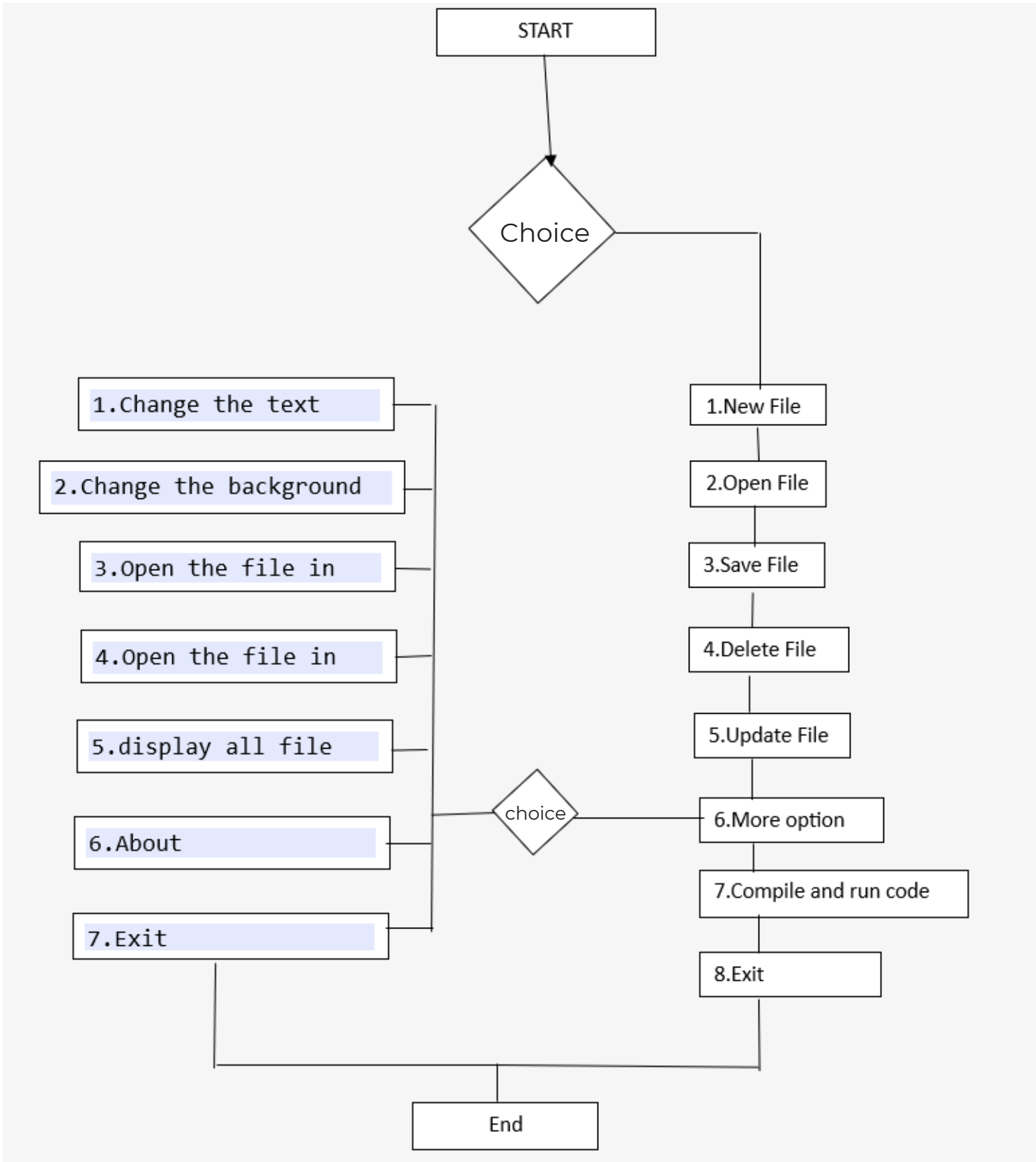
- **Inheritance:** Utilizing inheritance, I created a hierarchy of IDE elements, reducing redundancy in the code.

- **Polymorphism:** Polymorphism allowed for diverse code recognition challenges and IDEplay mechanics while maintaining code simplicity.

This approach, combined with c++ provided a solid foundation for IDE ensuring code organization, reusability, and adaptability for future enhancements.

DATA FLOW DIAGRAM




START
Choice
1.Change the text
2.Change the background
3.Open the file in
4.Open the file in
5.display all file
6.About
7.Exit
choice
1.New File
2.Open File
3.Save File
4.Delete File
5.Update File
6.More option
7.Compile and run code
8.Exit
End

USER INTERFACE DESIGN

The user interface (UI) of IDE has been meticulously crafted to be user-friendly and accessible to individuals of all age groups. The design focuses on simplicity and intuitiveness, ensuring a seamless development experience for everyone.

**1. Main Menu:

- The main menu provides a straightforward starting point for developers. It features large, easily recognizable option that lead to various IDE modes, settings, and achievements. The text is clear and legible, making navigation effortless.

**2. IDE Screen:

- During IDEplay, the IDE screen maintains a clean and uncluttered appearance. The code image is prominently disdevelop at the center, with space for developer inputs. Hints and scoring information are presented clearly, ensuring developers can focus on the challenge at hand.

The user interface of IDE embraces a design philosophy that puts user accessibility at the forefront. It has been crafted with simplicity and ease of use in mind, making it welcoming and engaging for developers of all ages.

USER INTERFACE DESIGN

The user interface (UI) of IDE has been meticulously crafted to be user-friendly and accessible to individuals of all age groups. The design focuses on simplicity and intuitiveness, ensuring a seamless development experience for everyone.

**1. Main Menu:

- The main menu provides a straightforward starting point for developers. It features large, easily recognizable icons that lead to various IDE modes, settings, and achievements. The text is clear and legible, making navigation effortless.

**2. IDE Screen:

- During IDE's development, the IDE screen maintains a clean and uncluttered appearance. The code image is prominently disdevelop at the center, with space for developer inputs. Hints and scoring information are presented clearly, ensuring developers can focus on the challenge at hand.

The user interface of IDE embraces a design philosophy that puts user accessibility at the forefront. It has been crafted with simplicity and ease of use in mind, making it welcoming and engaging for all developers .

INTRODUCTION TO LANGUAGES

The development of IDE was made possible through a carefully chosen technology stack, which encompassed programming languages, design tools, version control systems, and integrated development environments (IDEs). This chapter provides an overview of the technologies employed in the creation of the IDE.

Programming Languages:

1. **c++: The primary programming language used for IDE development, known for its conciseness and compatibility with windows.

CODING STANDARDS OF LANGUAGE USED

In the development of IDE adherence to coding standards was a fundamental practice to maintain code quality, readability, and consistency. The following coding standards were applied to the languages used in the development process:

1. c++:

- In c++ we followed naming conventions that reflect the nature and purpose of variables, functions, and classes. This promotes code readability and understanding.
- We utilized consistent indentation and spacing to ensure code formatting consistency, making the codebase more accessible for developers.

The application of these coding standards served to maintain the codebase's quality and readability. By following best practices, we aimed to facilitate collaboration among developers, streamline debugging, and enhance the overall stability and maintainability of IDE

TESTING TECHNIQUES

The quality and reliability of IDE were of paramount importance, and thus, comprehensive testing techniques were employed to ensure a robust IDE. The following testing techniques and test plan were integral to the development process:

Testing Techniques:

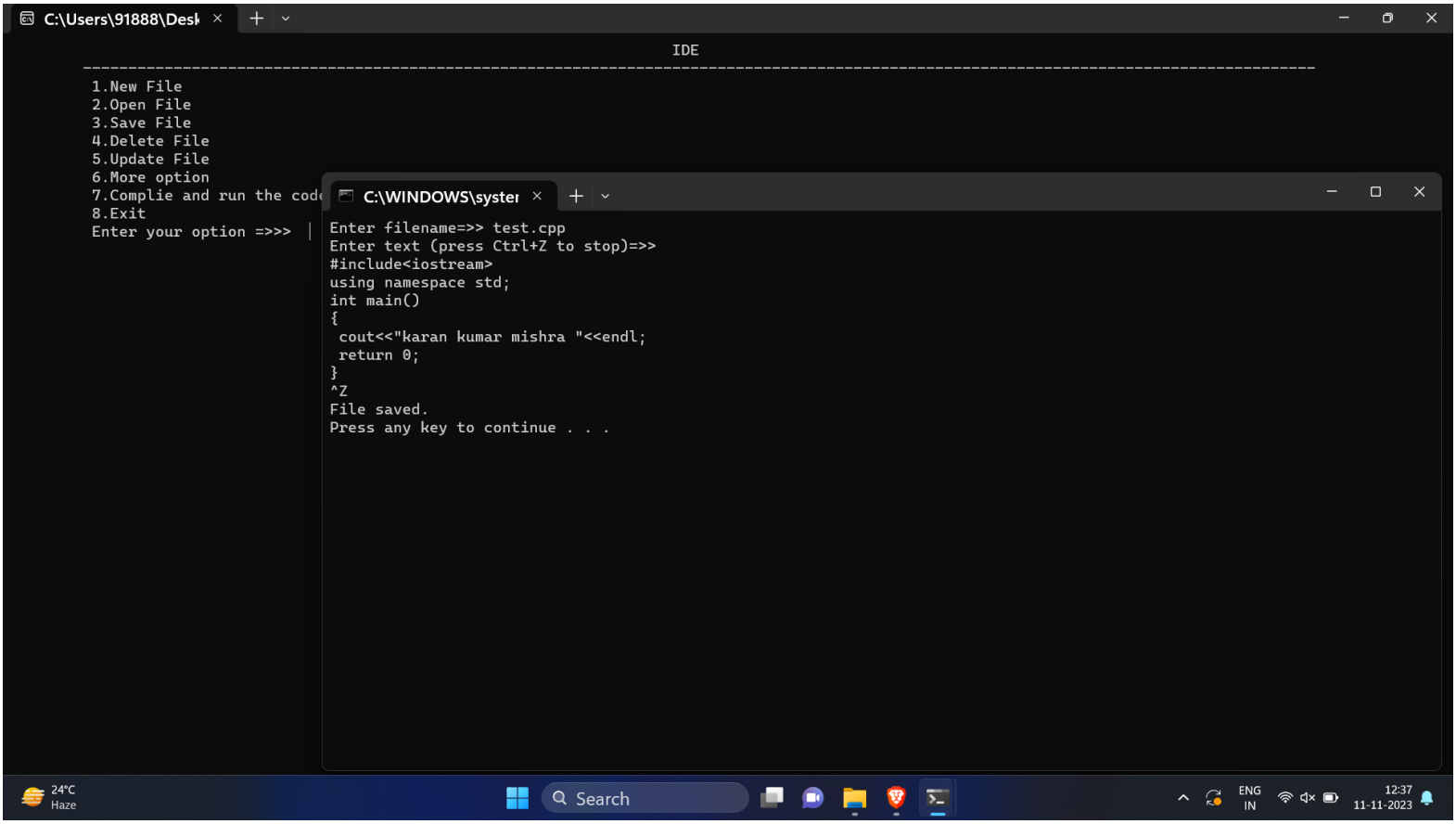
1. **Unit Testing:**
   - Individual IDE components were tested in isolation to verify that they functioned as expected. This ensured that each module, class, or method performed its intended purpose accurately.

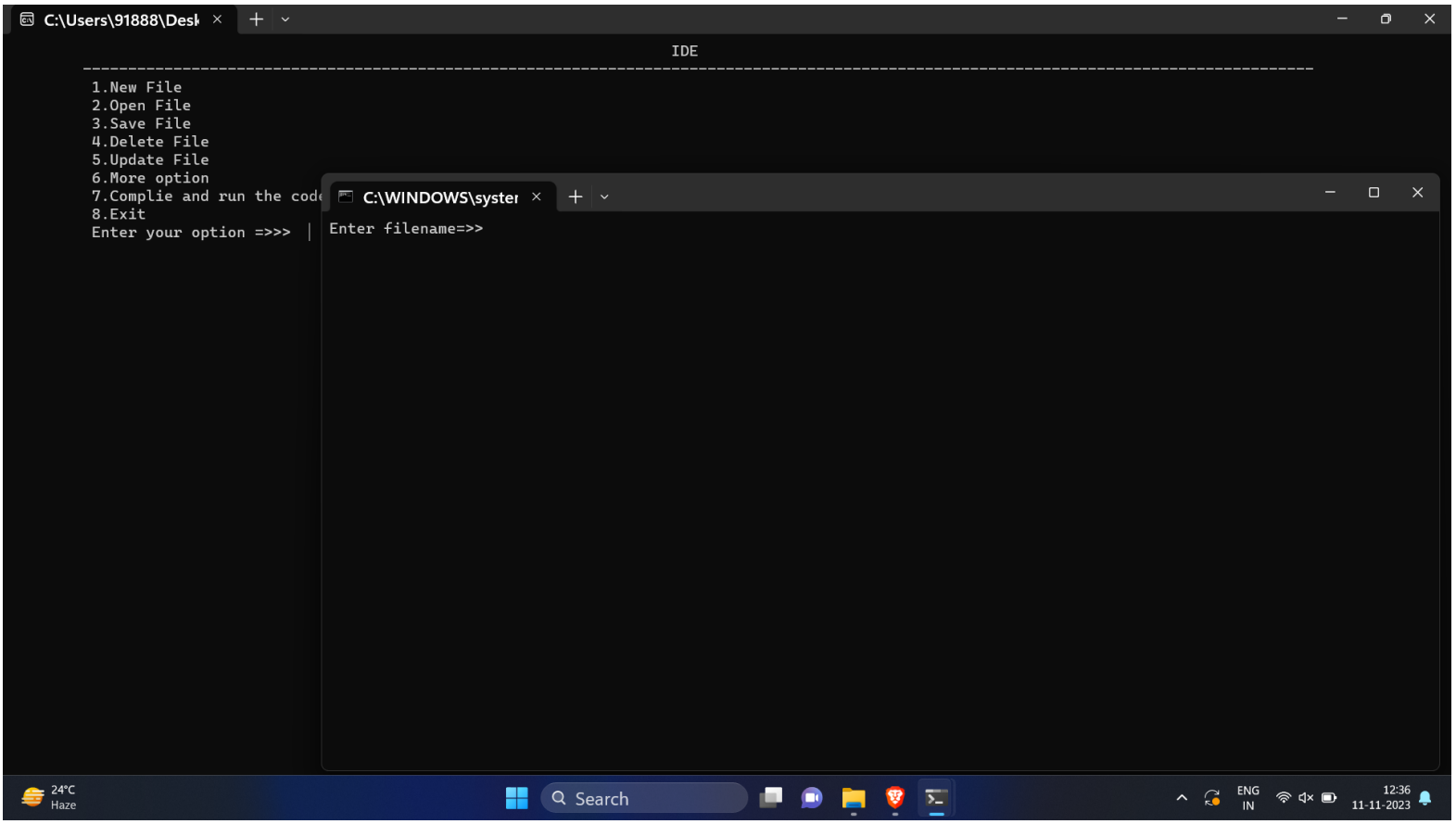
2. **Integration Testing:**
   - Integration tests were conducted to assess how various IDE components interacted with one another. This ensured that different parts of the IDE worked seamlessly when combined.

3. **User Testing:**
   - Real users were involved in testing to gain valuable insights into the IDE's usability and overall user experience. This feedback develop a crucial role in improving IDEplay and addressing user concerns.

4. **Performance Testing:**
   - Performance tests were conducted to assess the IDE's responsiveness, processing speed, and resource utilization. This ensured smooth IDEplay for developers across different devices and scenarios.

Chapter: 5 Results and Discussions





BACKEND REPRESENTATION (SNAPSHOT)