A
Mini-Project Report on

Unit Convertor

Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning by

Shubham Patil (22106047) Anup
Singh (22106104)
Mahesh Shinde(22106128)
Harsh Salunke (22106133)

Under the guidance of

Prof. Sayali Badhan

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai 2023-2024

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled “**Unit Convertor”** is a bonafide work of Shubham Patil (22106047), Anup Singh(22106104), Mahesh Shinde(22106128), Harsh Salunke(22106133) submitted to the University of Mumbai in partial fulfillment of the requiremen-t for the award of

Bachelor of Engineering in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

__________________	_________________
Prof. Sayali Badhan	Dr. Jaya Gupta
Mini Project Guide	Head of Department

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled "**Unit Converter**" by **Shubham Patil, Anup Singh ,Mahesh Shinde and Harsh Salunke** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2023-24***.-

External Examiner:

Internal Examiner:

Place: APSIT, Thane Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shubham Patil	Anup Singh	Mahesh Shinde	Harsh Salunke
(22106047)	(22106104)	(22106128)	(22106133)

ABSTRACT

In our increasingly interconnected world, the need to convert units of measurement is a common challenge. Whether it's converting currency while traveling, deciphering temperature scales, or converting between different units of length or weight, unit conversion is a fundamental requirement in various domains, including science, engineering, finance, and everyday life.

Index

CHAPTER 1 INTRODUCTION

1. INTRODUCTION

In our increasingly interconnected world, the need to convert units of measurement is a common challenge. Whether it's converting currency while traveling, deciphering temperature scales, or converting between different units of length or weight, unit conversion is a fundamental requirement in various domains, including science, engineering, finance, and everyday life.

To address this need, we have undertaken the development of a Unit Converter Application. This application aims to simplify unit conversions by providing an easy-to-use, efficient, and accurate tool for

users to convert values from one unit to another seamlessly.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

1. Introduction

Unit conversion is a common computational task across various fields, and developing a Unit Converter in Java presents a valuable opportunity to explore practical Java application development. This literature survey aims to provide an overview of important concepts, technologies, and best practices related to creating a unit converter in the Java programming language.

2. Java Programming for Application Development

Understanding the relevance of Java for application development is essential before diving into the specifics of unit converters.

Java's Versatility: Java's platform independence, object-oriented nature, and robust libraries make it a popular choice for developing cross-platform applications.

Graphical User Interface (GUI): Java offers versatile GUI frameworks such as Swing and JavaFX, which enable the creation of user-friendly and interactive graphical interfaces.

3. Unit Conversion Concepts

Developing an effective unit converter requires a clear grasp of unit conversion concepts. Here, we discuss the foundational mathematical and logical principles behind unit conversions.

Ratio and Proportion The fundamentals of unit conversion often rely on ratios and proportions to establish relationships between different units.

Dimensional Analysis: Understanding dimensional analysis and its application in solving complex unit conversion problems.

4. Java GUI Frameworks

Efficient unit conversion applications require intuitive graphical interfaces. This section explores the choice of Java GUI frameworks for creating user-friendly interfaces.

Java Swing and JavaFX: A comparison of Java Swing and JavaFX in terms of their capabilities for building responsive and visually appealing GUIs for unit converters.

User Interface Design:Discussion of best practices for designing intuitive user interfaces using Java GUI components.

5. Implementation Techniques

The practical implementation of a unit converter in Java involves various techniques, including input validation, unit categorization, and conversion algorithms.

Input Validation: Strategies for efficient user input validation and error handling in Java applications.

Unit Organization: Modular approaches to categorizing and organizing units for seamless conversion within the application.

Conversion Algorithms:Presentation of algorithms for converting units both within and between different categories, emphasizing computational efficiency.

6. Internationalization and Localization

A versatile unit converter should support multiple languages and regional preferences. This section touches upon internationalization and localization techniques in Java.

ResourceBundle Class:An explanation of Java's ResourceBundle class and its role in creating multilingual unit converter applications.

Locale-Based Adaptation: Strategies for adapting the user interface and formatting output based on the user's locale.

CHAPTER 3

Problem Statement

In today's interconnected world, the need for a versatile and user-friendly unit conversion tool is increasingly evident. Individuals across various fields and industries often encounter the challenge of converting units of measurement for diverse purposes, including scientific calculations, engineering tasks, financial transactions, and everyday activities. However, existing unit conversion solutions often lack user-friendliness, flexibility, and cross-platform compatibility.

The problem to be addressed in this project is the absence of a comprehensive and easily accessible unit converter that caters to a wide range of unit categories, is adaptable to user preferences, and provides a seamless user experience. Current unit conversion tools may lack critical features, such as support for international units, user-defined units, and cross-device compatibility, making them less than ideal for modern users.

Furthermore, many existing unit conversion applications may not offer the level of

customization and localization required to meet the diverse needs of users in different regions and industries. Consequently, there is a need for a robust and user-centric unit converter application developed in Java that overcomes these limitations and provides a versatile solution for unit conversion tasks.

This project aims to address these challenges by developing a Java-based Unit Converter application that is user-friendly, adaptable, and capable of handling a wide range of unit categories, all while maintaining an intuitive and visually appealing user interface. The application will also prioritize internationalization and localization features to ensure it can be used effectively by individuals worldwide. By creating this comprehensive Unit Converter in Java, the project seeks to offer a versatile solution to the unit conversion needs of users across diverse fields and backgrounds.

CHAPTER 4

Experimental Setup

3. Experimental Setup

4.1 Hardware Setup

The hardware requirements for your "Unit Converter in Java" project are relatively modest since Java is a cross-platform language known for its "Write Once, Run Anywhere" capability. Below are the basic hardware requirements for developing and running a Java-based application:

1. Computer: You will need a personal computer to develop the Java application. Most modern computers, whether running Windows, macOS, or Linux, will suffice.

2. Processor:A multi-core processor (e.g., Intel Core i5 or AMD Ryzen) will provide a smoother development experience

3. RAM A minimum of 4GB of RAM is recommended for running the development environment and your Java application effectively. More RAM can improve performance, especially with larger projects.

4. Storage: A standard SSD (Solid State Drive) or HDD (Hard Disk Drive) with sufficient storage capacity (at least 250GB) for your project files, development tools, and any libraries or dependencies is necessary.

5. Display:A monitor or screen with a resolution of 1920x1080 or higher will make coding and testingmore comfortable. A dual-monitor setup can be beneficial for productivity but is not a requirement.

6. Keyboard and Mouse: A standard keyboard and mouse for input are essential.

7. Internet Connection: A stable internet connection is required for downloading development tools, libraries, and dependencies, as well as for accessing online resources and documentation.

4.2 Software Setup

The software requirements for your "Unit Converter in Java" project include the development tools, libraries, and dependencies necessary to create, build, and run your Java application. Here are the essential software requirements:

1. Java Development Kit (JDK) we will need a Java Development Kit installed on your computer. Javais the programming language you'll use to develop your unit converter. You can download the latest JDK from the official Oracle website or use an open-source alternative like OpenJDK.

2. Integrated Development Environment (IDE):An Integrated Development Environment provides a user-friendly interface for writing, testing, and debugging Java code. Some popular Java IDEs include:

Eclipse: A free, open-source IDE with a large user base and a wide range of plugins.

JavaFX : If you plan to create a graphical user interface (GUI) for your unit converter, JavaFX is a popular choice. JavaFX is included with most recent Java distributions, but you might need to install it separately depending on your JDK version.

8. User Interface Design Tools : If you're creating a GUI, you might need graphic design software like Adobe XD, Sketch, or Figma for designing the user interface.

9. Documentation Tools : For project documentation, you can use tools like Microsoft Word, LaTeX, or Markdown editors (e.g., Visual Studio Code with Markdown extensions).

The specific software requirements may vary depending on your project's complexity and goals. Ensure that you have the necessary tools and dependencies installed to develop, test, and deploy your unit converter effectively. Additionally, always keep your software up to date to benefit from security patches and the latest features.

CHAPTER 5

Proposed System & Implementation

CHAPTER 6 Conclusion

Conclusion

This literature survey has provided an introductory exploration of the fundamental concepts, tools, and techniques relevant to building a unit converter in Java. By drawing on this foundational knowledge and considering best practices documented in practical Java application development, you can embark on the creation of an efficient and user-friendly unit converter application.

References