Python Project
On
CovertEase
(All-in-one-scientific converter)

Submission date:November 10th,2024
Course Code:CSM216

Git-hub link,
https://github.com/Dakshin2k05/ConvertEase

Submitted by,
Name: Dakshineswar.M
Reg.no.:12303280
Roll no.:RK23CHA20

Submitted to,
Name: Aman Kumar
Reg.no.:63642

In partial fulfilment for the requirements of the award of the degree of
"B. Tech CSE Data Science and Machine Learning"

"School of Computer Science and Engineering"
Lovely Professional University
Phagwara, Punjab

ACKNOWLEDMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the development of my project using Tkinter in Python.

First and foremost, I am deeply thankful to my teacher,Mr.Aman Kumar, for their invaluable guidance, encouragement, and expertise, which greatly contributed to the completion of this project.

I extend my heartfelt thanks to my friends for their collaboration, support, and constructive feedback during various stages of the project. Their insights and suggestions were instrumental in overcoming challenges and enhancing the final output.

Additionally, I am grateful to my institution, Lovely Professional University and Upgrad for providing the resources and platform to explore and implement this project. I also acknowledge the role of documentation in resolving technical issues and expanding my understanding of Tkinter and Python.

Lastly, I thank my family and friends for their constant support and encouragement, which motivated me to achieve my goals.

Thank you all for your contributions to the success of this project.

Table of Content

1.INTRODUCTION

The "**ConvertEase**" is a Python-based application designed to consolidate a wide variety of conversion functionalities into a single, intuitive platform. In an era where digital tools are increasingly relied upon for calculations and data processing, users often face the inconvenience of using multiple separate tools for different types of conversions. This project aims to address that issue by providing a unified solution capable of handling diverse conversion tasks with ease and efficiency.

The application is built using the **Tkinter library**, which allows for the creation of a graphical user interface (GUI) that is simple, user-friendly, and visually appealing. By integrating multiple conversion functions—such as currency conversion, temperature conversion, digital storage calculations, and even advanced tools like encryption/decryption—the application caters to a wide range of user needs. Each feature is implemented as an independent module, ensuring modularity and scalability in the application's design.

The primary purpose of this project is to simplify the process of conversions for users in fields like finance, science, engineering, and day-to-day tasks. For example:

- A student or researcher can quickly convert temperatures or calculate fuel economy.
- A developer or IT professional can encrypt/decrypt data or compute digital storage conversions.
- A traveler can access live currency conversion rates to assist with financial planning.

Additionally, this project showcases the potential of Python and its ecosystem of libraries in solving real-world problems efficiently. By using libraries like Requests for live data fetching, and **Pillow** for graphical enhancements, the project emphasizes Python's versatility and suitability for building robust, feature-rich applications.

The significance of the **"ConvertEase"** lies not just in its functionality but also in its educational value, as it serves as an example of effective software design and implementation. The project highlights the following key aspects:

1. **Problem-Solving Approach**:Addresses the need for a centralized tool for diverse conversions.
2. **Usability**:Designed for users of all technical backgrounds, ensuring accessibility.
3. **Efficiency:**Provides accurate results with minimal input from the user.
4. **Extensibility:** The modular design allows for future additions and improvements.

In conclusion, the "**ConvertEase** " is more than just a utility application—it is a demonstration of the power of Python programming in creating solutions that combine functionality, simplicity, and efficiency. It is aimed at reducing the effort and complexity involved in performing routine and specialized conversions, making it a valuable tool for users across different domains.

2.OBJECTIVE & SCOPE

Objectives:

The "**ConvertEase**" project is designed with the following key objectives:

1. **Centralized Conversion Platform:**
   To create a unified tool that combines multiple types of unit conversions, eliminating the need for separate applications for individual conversion tasks.
2. **Ease of Use:**
   To provide an intuitive, user-friendly graphical interface using Python's Tkinter library, ensuring that users with minimal technical expertise can use the application effortlessly.
3. **Accuracy and Reliability:**
   To ensure that all conversions are performed with high precision and verified using standard formulas or live data, such as real-time currency exchange rates.
4. **Versatility:**
   To support a wide variety of conversion categories, including but not limited to currency, temperature, weight, volume, speed, time, energy, and digital storage units.
5. **Extensibility:**
   To design the application with modularity in mind, allowing for easy integration of additional conversion categories and features in the future.
6. **Educational and Practical Value:**
   To serve as a practical tool for everyday use and as an example of effective Python programming, showcasing the integration of GUI design and backend functionality.

Scope:

The scope of the "**ConvertEase**" encompasses the following aspects:

1. **Wide Range of Conversions**
   The application includes support for various conversions, such as:
   - **Currency Converter** (real-time exchange rates).
   - **Temperature Converter** (Celsius, Fahrenheit, Kelvin, etc.).
   - **Weight and Mass Converter** (kilograms, pounds, grams, etc.).
   - **Volume Converter** (liters, gallons, cubic meters, etc.).
   - **Speed Converter** (miles per hour, kilometers per hour, meters per second, etc.).
   - **Time Converter** (seconds, minutes, hours, days, etc.).
   - **Digital Storage Converter** (bytes, kilobytes, megabytes, etc.).
   - **Plane Angle Converter** (degrees, radians, gradians, etc.).
   - **Energy Converter** (joules, calories, kilowatt-hours, etc.).
2. **Advanced Functionalities**
   - **Fuel Economy Calculator:** To compute fuel efficiency based on distance traveled and fuel consumed.
   - **Encryption/Decryption Tool:** To provide a secure way to encode and decode text or data.
   - **Base Converter:** For converting numbers between various bases (binary, decimal, hexadecimal, etc.).
3. **User Interface and Experience**
   - A simple and responsive GUI with clear navigation and labeled buttons for each converter type.
   - Error handling for invalid inputs to prevent application crashes.
   - A centralized main menu for quick access to all features.
4. **Technical Specifications**
   - Developed using **Python** programming language.
   - The GUI is built using **Tkinter**, ensuring platform compatibility and simplicity.

- Real-time data retrieval for currency exchange rates using the **Requests** library.

5.**Target Audience**

- Students and researchers needing quick and accurate unit conversions.
- Professionals in finance, IT, and engineering fields.
- General users requiring a single platform for diverse conversion needs.

6.**Limitations**

- The application relies on an active internet connection for live currency rates.
- Precision of certain conversions (e.g., currency) may vary slightly due to rounding or data source constraints.

The **"ConvertEase"** is designed to be a practical, reliable, and user-centric tool that meets the diverse conversion needs of its target audience while offering room for future growth and enhancement.

3. APPLICATION TOOLS

The development of the **"ConvertEase"** project involved the use of various tools, software, and libraries to ensure functionality, usability, and efficiency. Below is a list of the application tools used:

1.Programming Language:

- **Python**: The primary language used to develop the application, known for its simplicity, versatility, and extensive library support.

2.Integrated Development Environment (IDE):

- **Visual Studio Code:** Used for writing, debugging, and managing the project's codebase.
- **PyCharm:** Occasionally used for advanced debugging and project structuring.

3.Libraries/Packages:

- **Tkinter:** The main library used to design and implement the graphical user interface (GUI), ensuring simplicity and user-friendliness.
- **Requests:** Used to fetch live data for features such as real-time currency exchange rates.
- **Pillow:** Used to enhance the visual elements of the application, such as icons and images.

4.Design Tools:

- **Figma:** Used for designing the application's user interface mockups and visualizing the layout of the GUI before implementation.

5.Version Control:

- **Git:** Employed for version control to manage the project's codebase, enabling collaboration and tracking of changes.

4.Project Design

The "ConvertEase" project follows a modular and organized structure, combining the backend logic for computations with a user-friendly graphical user interface (GUI) using Python's Tkinter library. Below is a detailed breakdown of the project's design:

1.Application Structure:

The project consists of the following core components:

⇒ **Main Menu**:

- Acts as the central hub of the application, presenting all available converters in a categorized button-based layout.
- Includes a "Logout" button to close the application.

⇒ **Conversion Modules**:

Each conversion module is designed as an independent function or class, making the code modular and easier to maintain. The modules include:

- Currency Converter
- Temperature Converter
- Weight and Mass Converter
- Volume Converter
- Speed Converter
- Time Converter
- Energy Converter
- Digital Storage Converter
- Fuel Economy Calculator
- Plane Angle Converter
- Encryption/Decryption Tool
- Base Converter
- Area Converter

⇒ **GUI Layout**:

- The GUI is structured using Tkinter's frames, buttons, and labels, ensuring a clear and consistent design across all modules.
- Each module has its own dedicated frame for displaying conversion input fields, dropdowns, and results.

2.Core Functionalities:

- **Input Handling:**
  Input fields accept user data for conversion and validate it to ensure that only numeric or valid text inputs are processed.
- **Calculation Logic**:
  Each conversion module contains its own logic, implemented using Python functions.

3.Interaction Flow:

⇒**Main Menu Navigation**:

- Users select the desired conversion module from the main menu, which navigates them to the respective conversion screen.
- After completing the conversion, users can return to the main menu.

⇒**Conversion Execution**:

- Users provide inputs in the designated fields, select relevant units from dropdowns, and click the "Convert" button to display results.

⇒**Logout**:

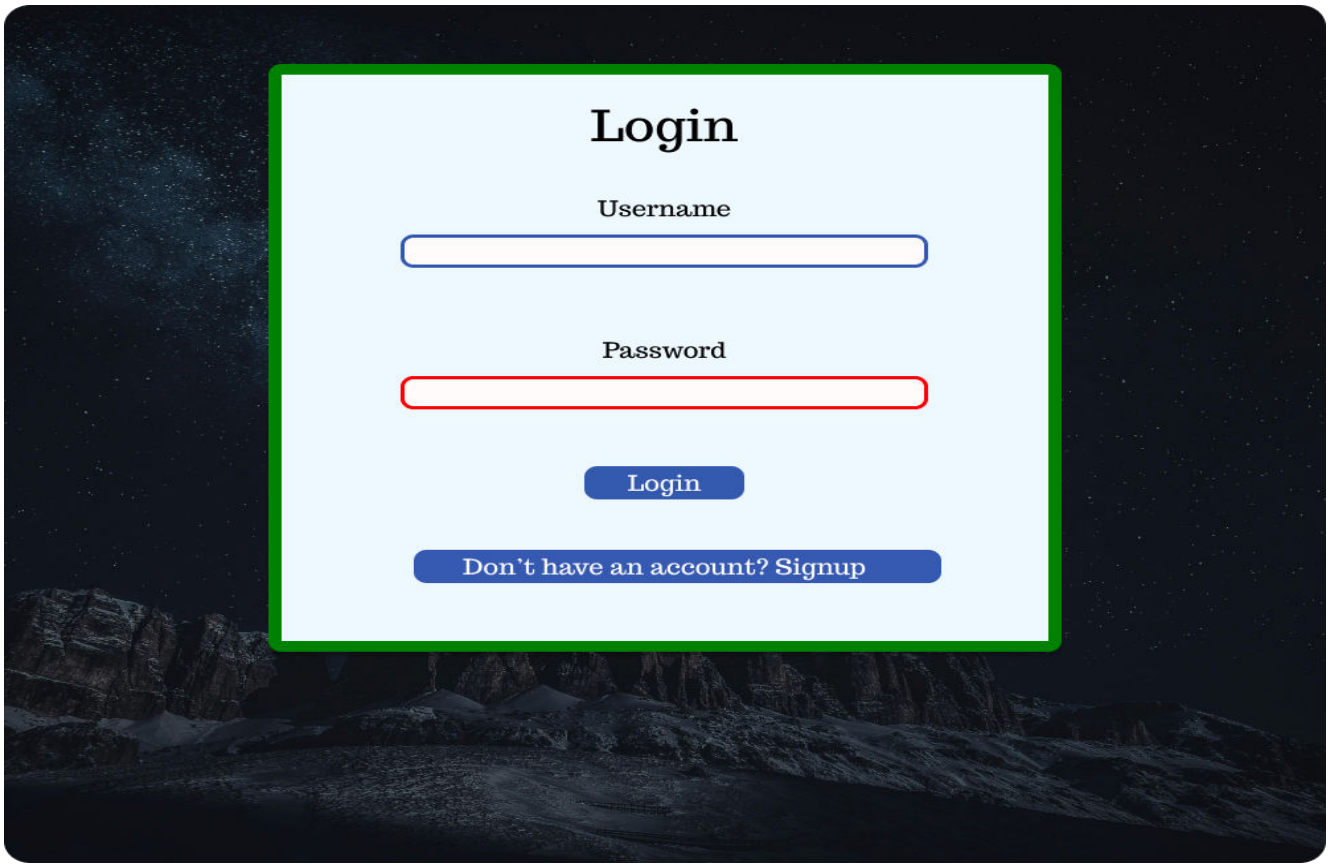
- The "Logout" button terminates the application gracefully.

4.Graphical User Interface (GUI) Design:

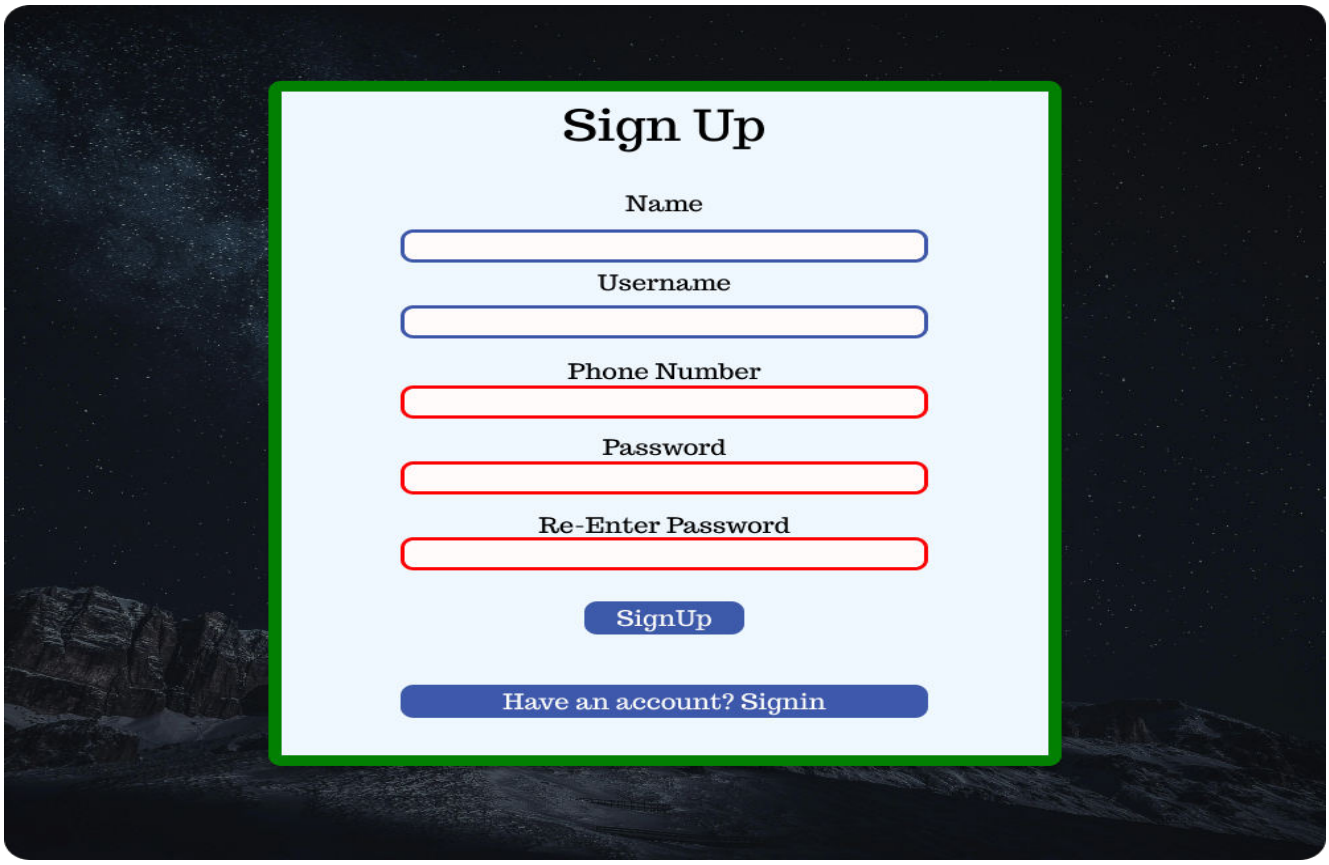
- The application's interface design was initially created using **Figma**, ensuring a clean and user-friendly layout.
- Figma mockups helped structure the placement of buttons, labels, and menus, providing a visual reference for the development process.
- This design workflow enabled a smooth transition from concept to implementation using Tkinter, ensuring consistency between the design and final application.
- Some of the figma designs included below

Overall design of ***ConvertEase***

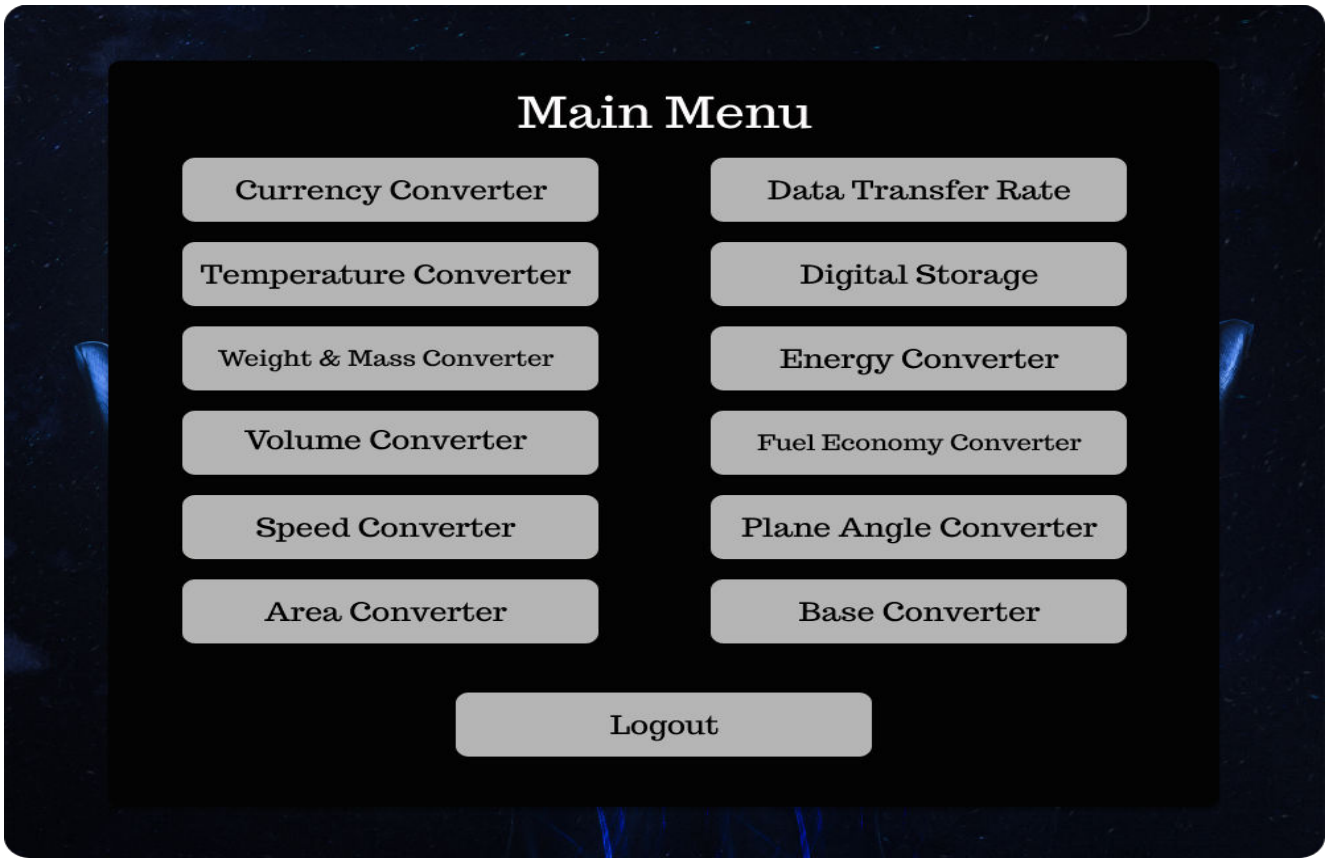
Getting Started Page Design



Login Page Design

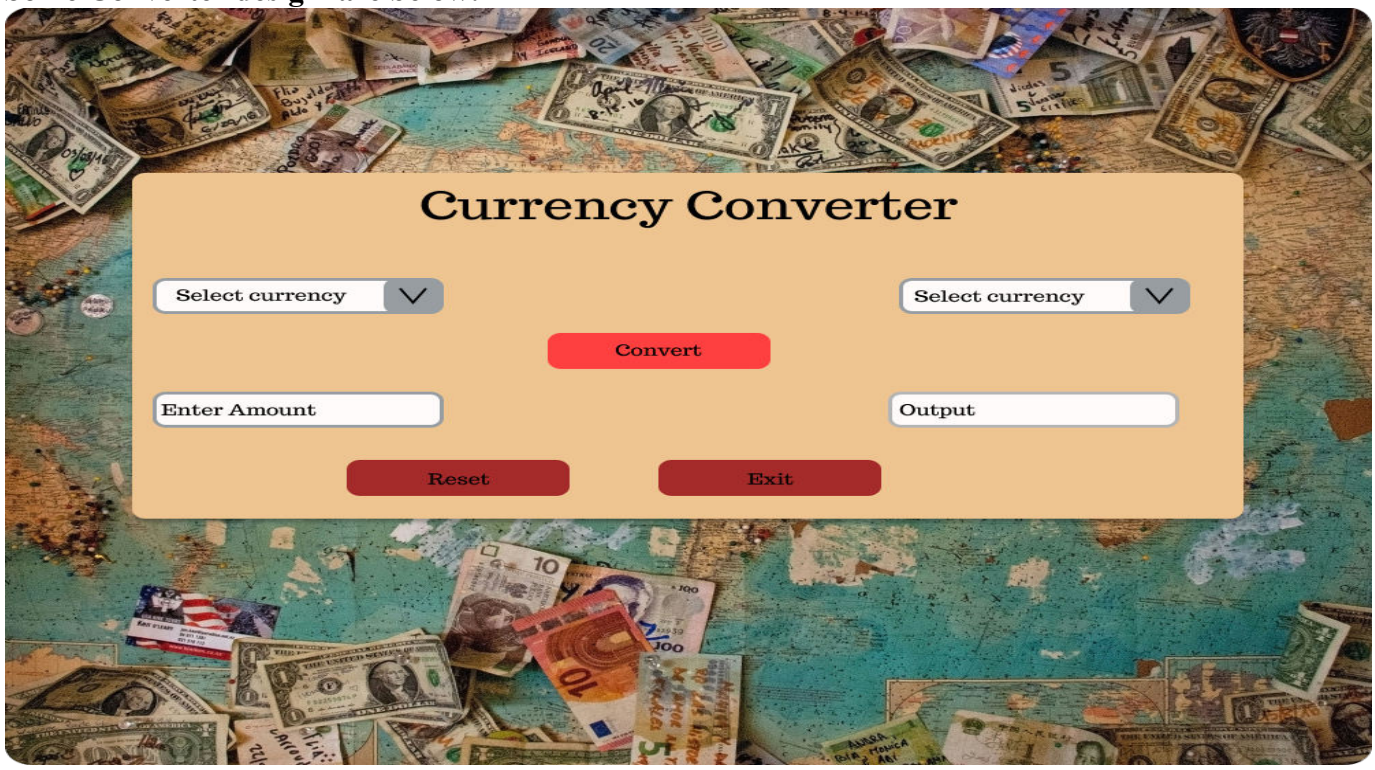


Sign Up Page Design

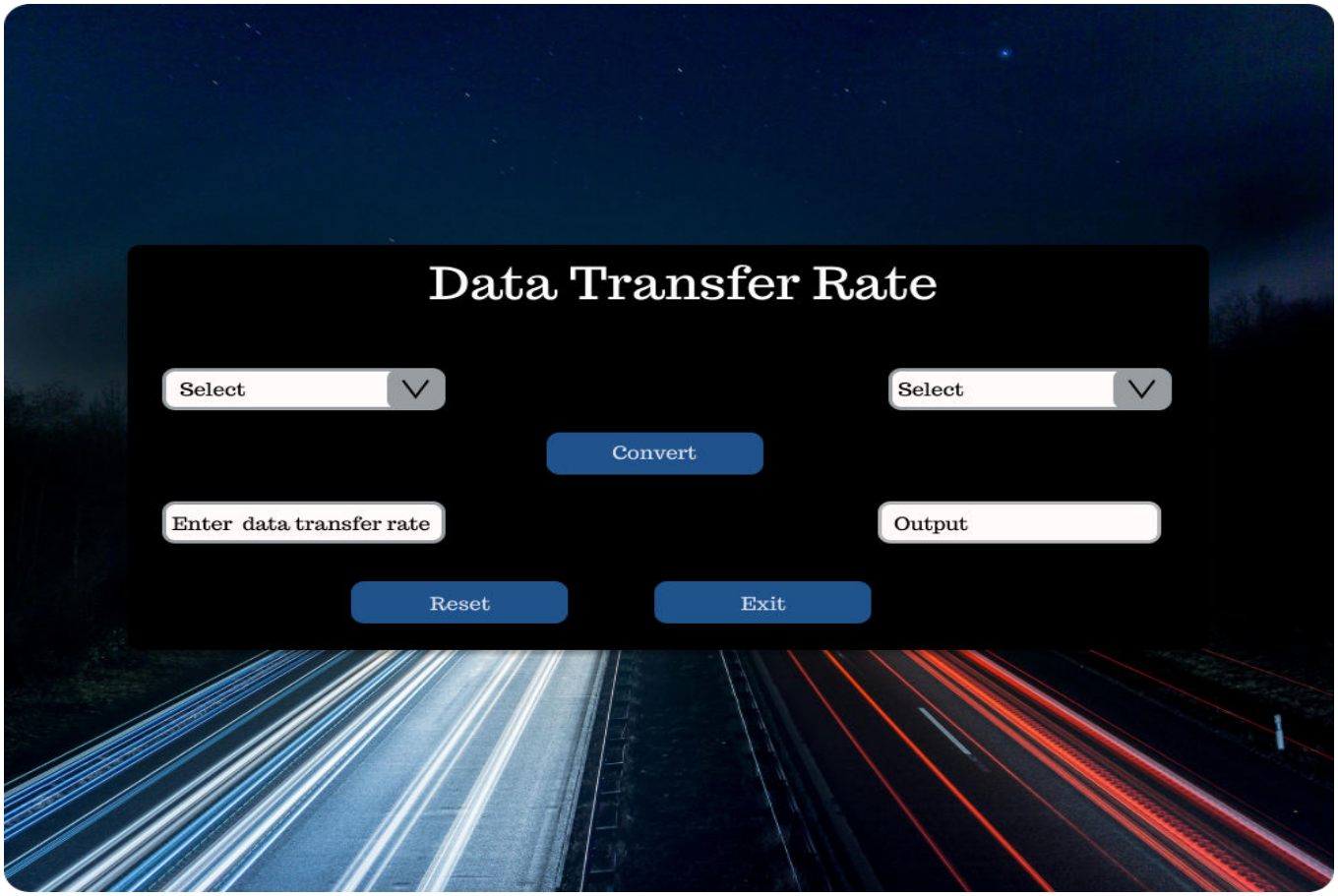


Main Menu Design

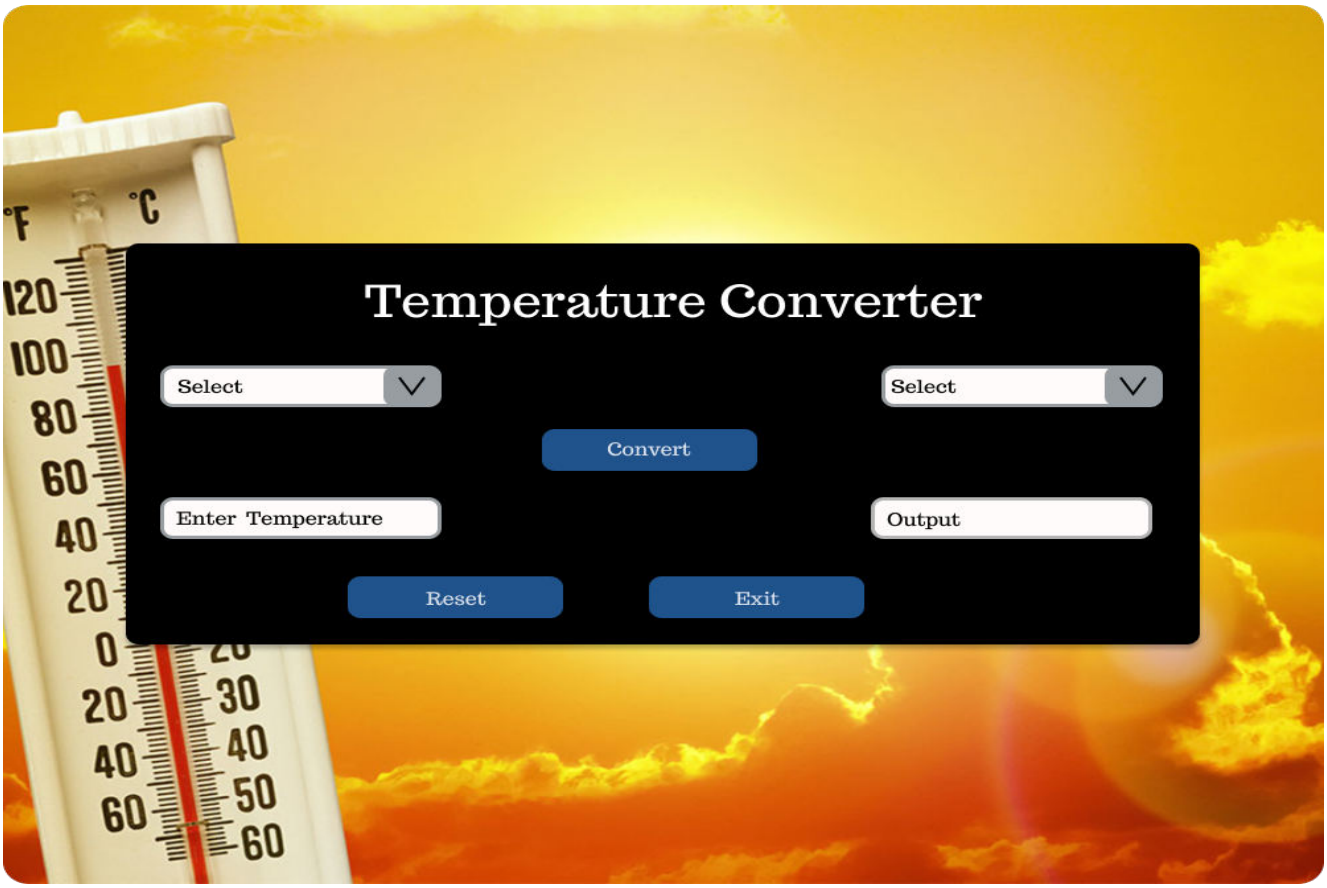
Some Converter design are below:



Currency Converter Page Design



Data Transfer Rate Page Design



Temperature Converter Design

Digital Storage unit design page

5.Modular Design

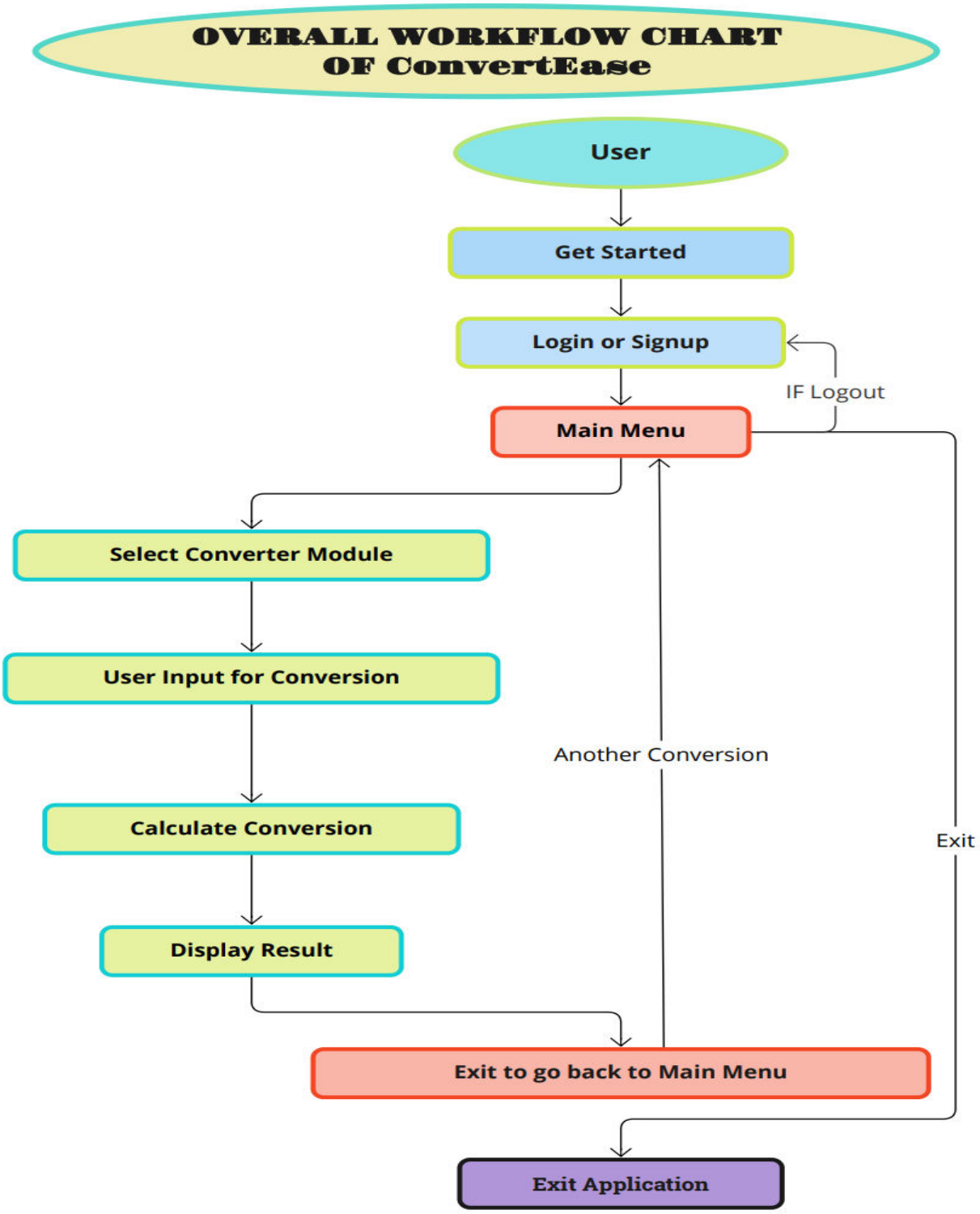
- Each converter is encapsulated in its own function or class,
- owing easy debugging and future enhancements.
- The main menu dynamically loads modules, ensuring smooth transitions between functionalities.

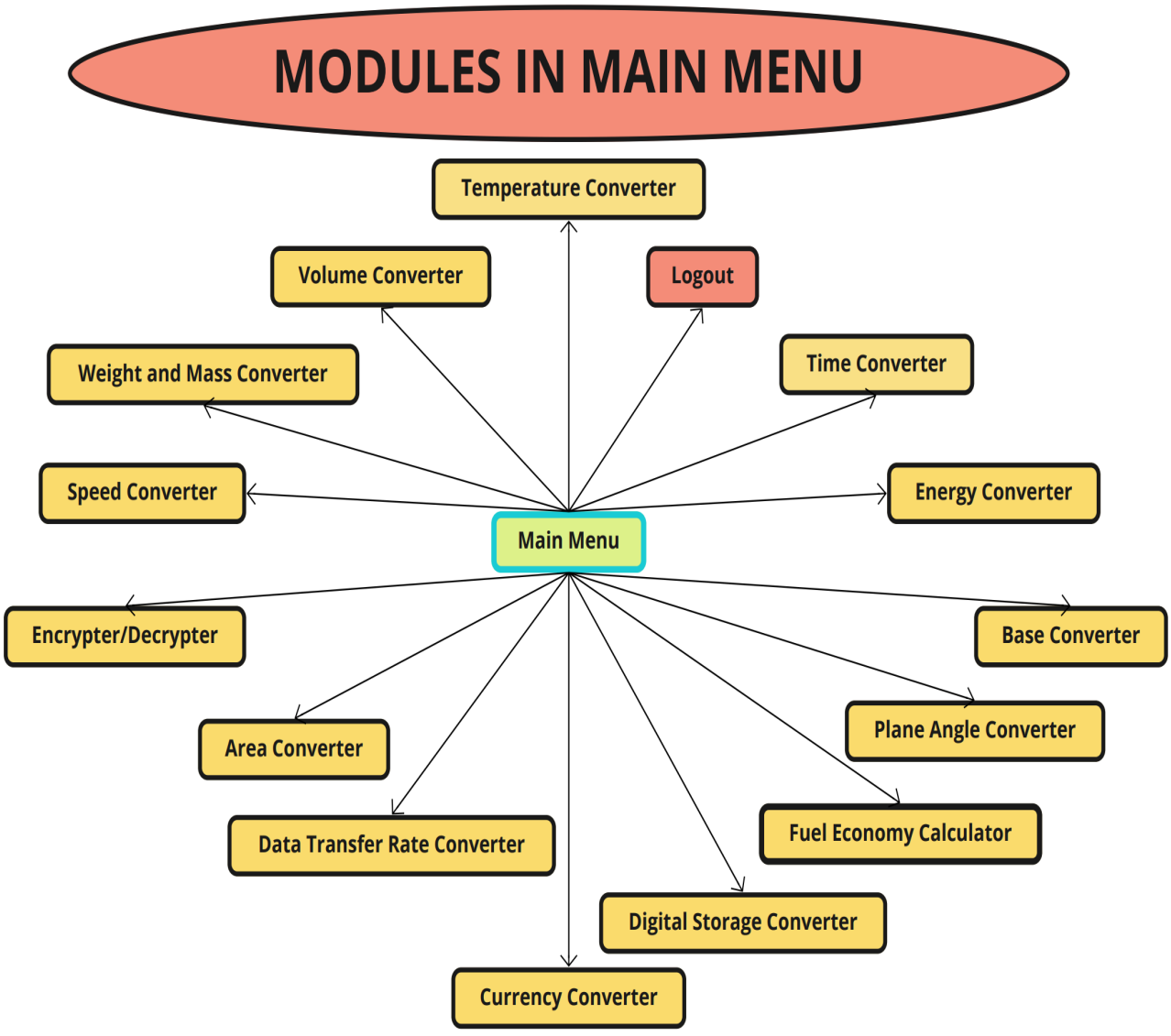
6.Reusability

- Shared components (e.g., input fields, dropdown menus, and result labels) are implemented as reusable functions to reduce redundancy.

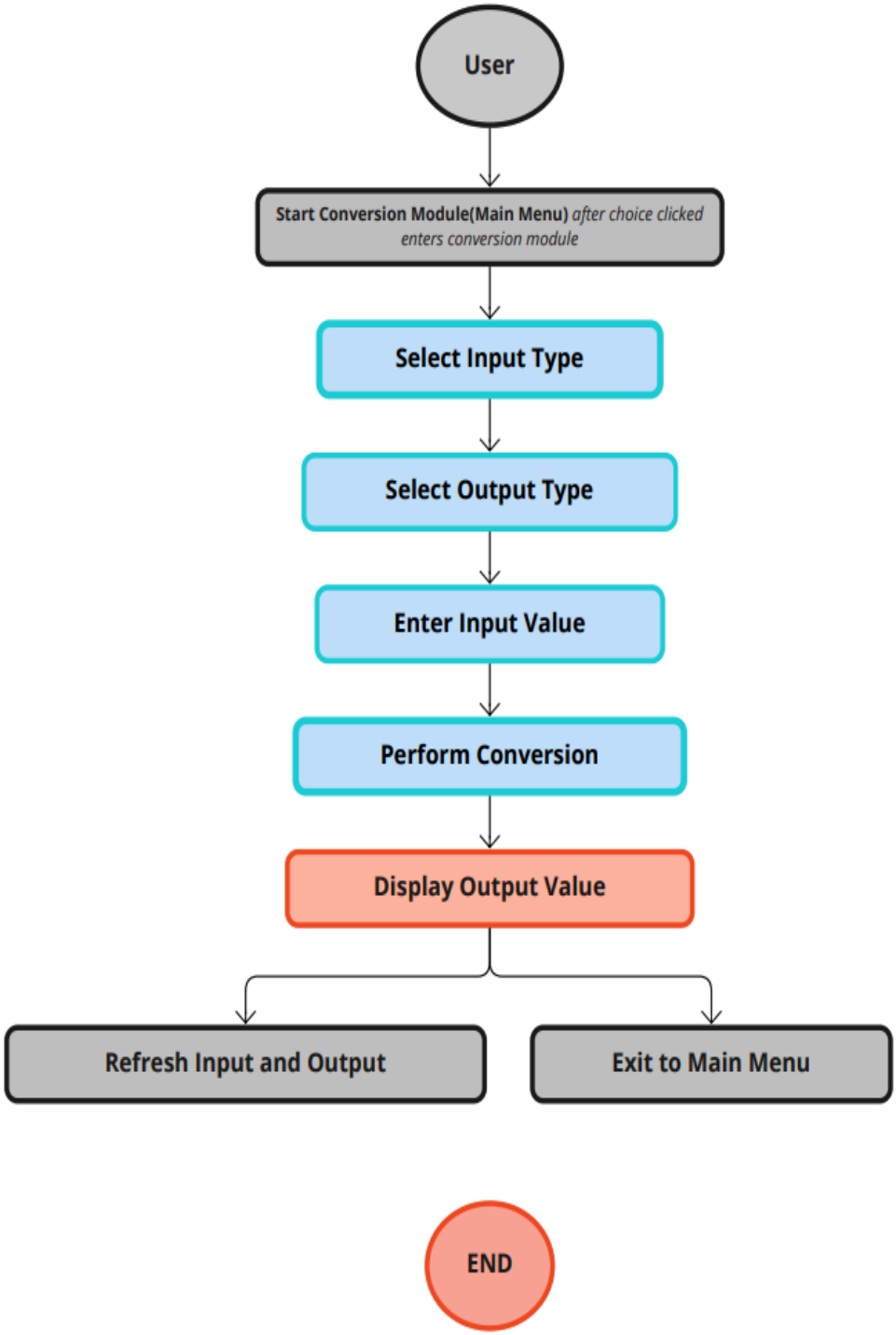
The project design ensures modularity, simplicity, and ease of use, making the **"ConvertEase"** a robust and scalable application.

5. FLOWCHART


OVERALL WORKFLOW CHART
OF ConvertEase
User
Get Started
Login or Signup
IF Logout
Main Menu
Select Converter Module
User Input for Conversion
Calculate Conversion
Display Result
Another Conversion
Exit
Exit to go back to Main Menu
Exit Application

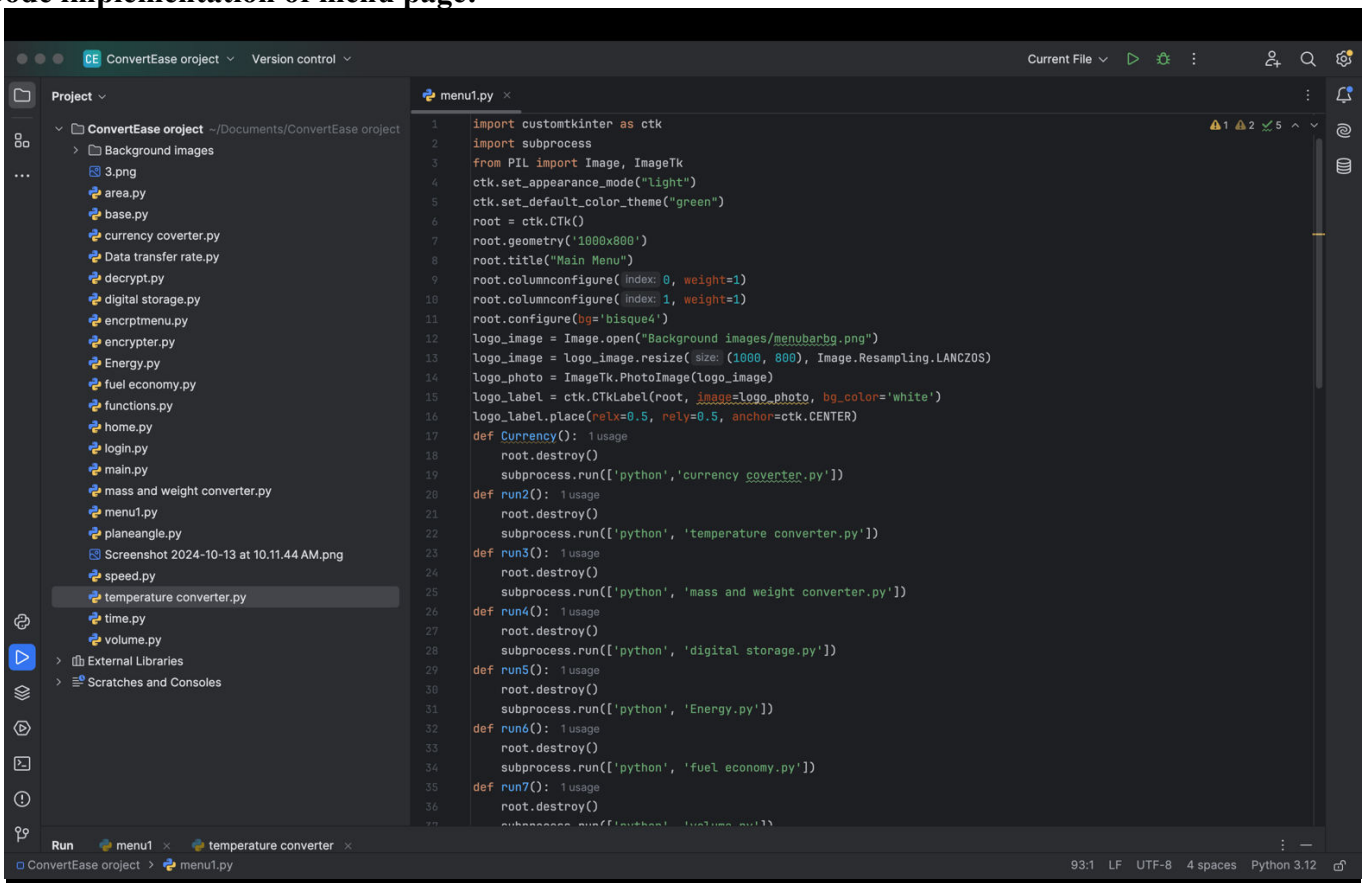

MODULES IN MAIN MENU
Temperature Converter
Volume Converter
Logout
Weight and Mass Converter
Time Converter
Speed Converter
Energy Converter
Main Menu
Encrypter/Decrypter
Base Converter
Area Converter
Plane Angle Converter
Data Transfer Rate Converter
Fuel Economy Calculator
Digital Storage Converter
Currency Converter

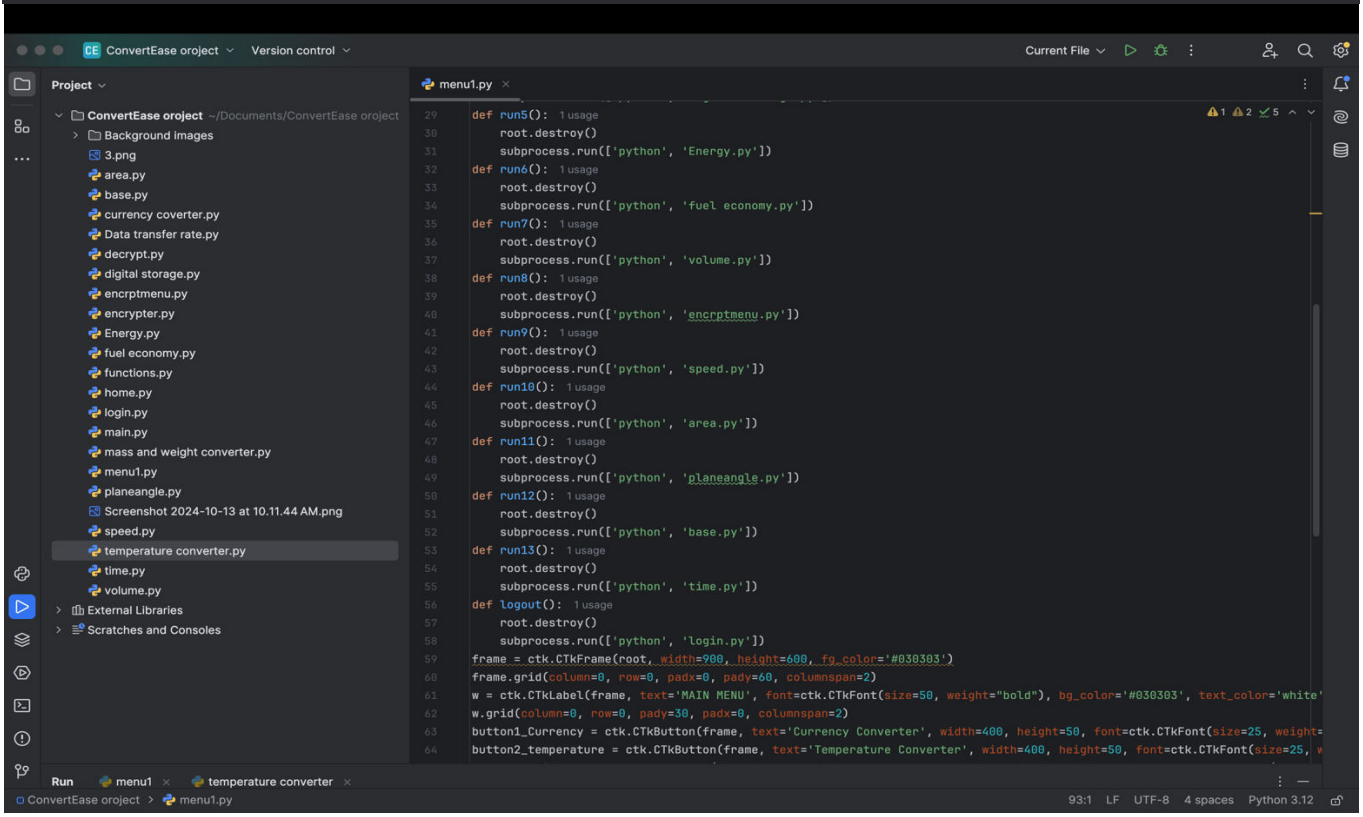
WORKFLOW OF EACH CONVERSION MODULE

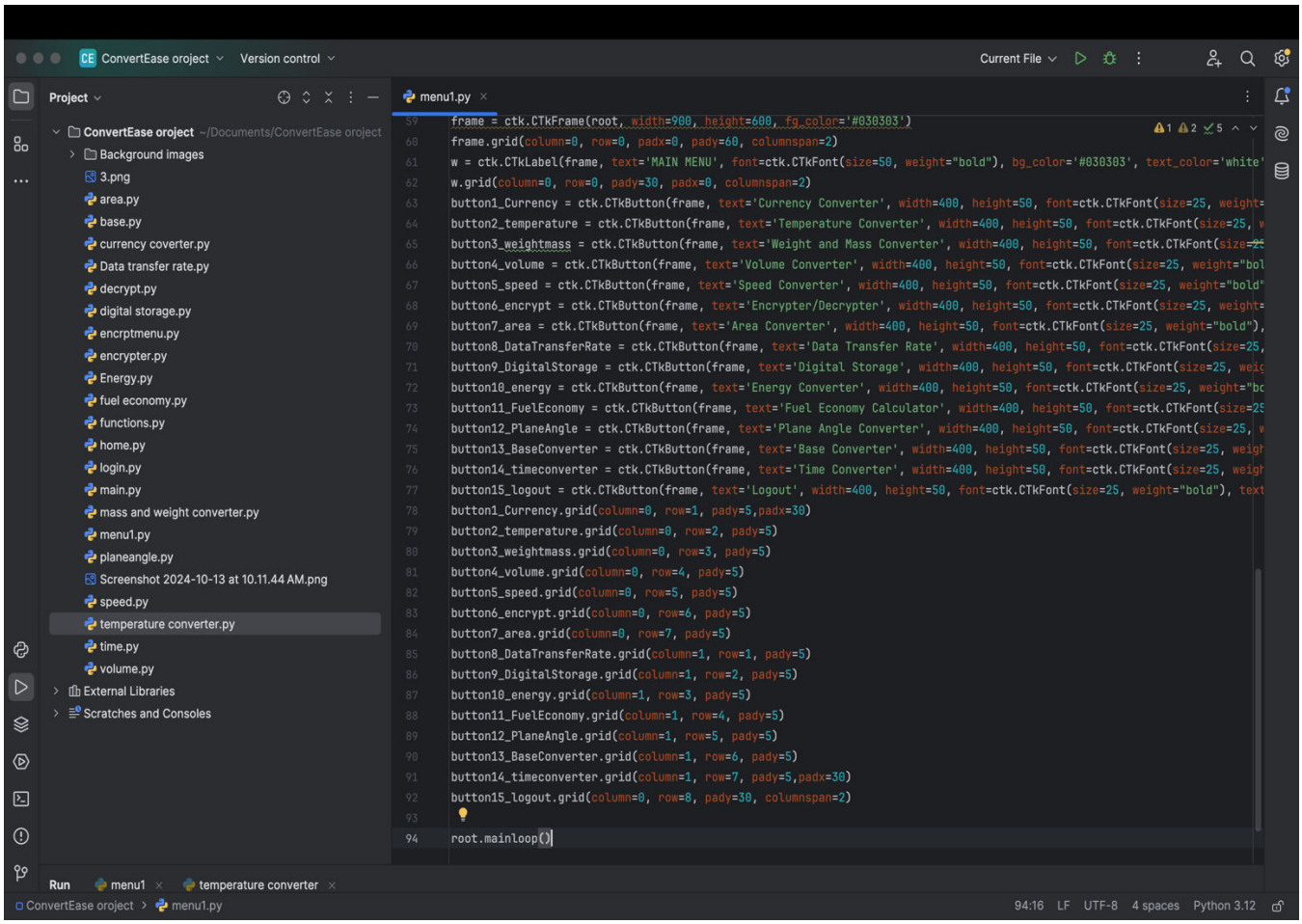


6. PROJECT IMPLEMENTATION

Code implementation of menu page:

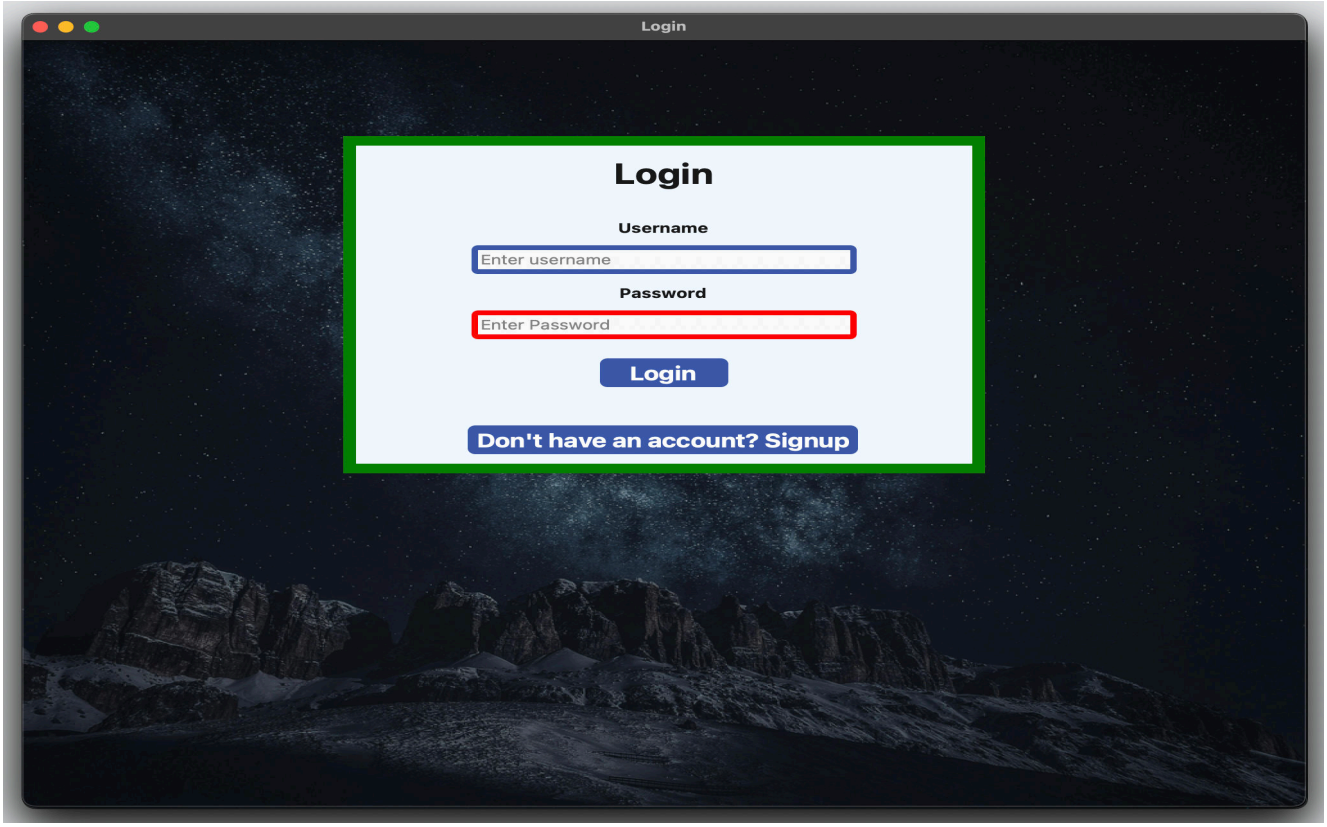




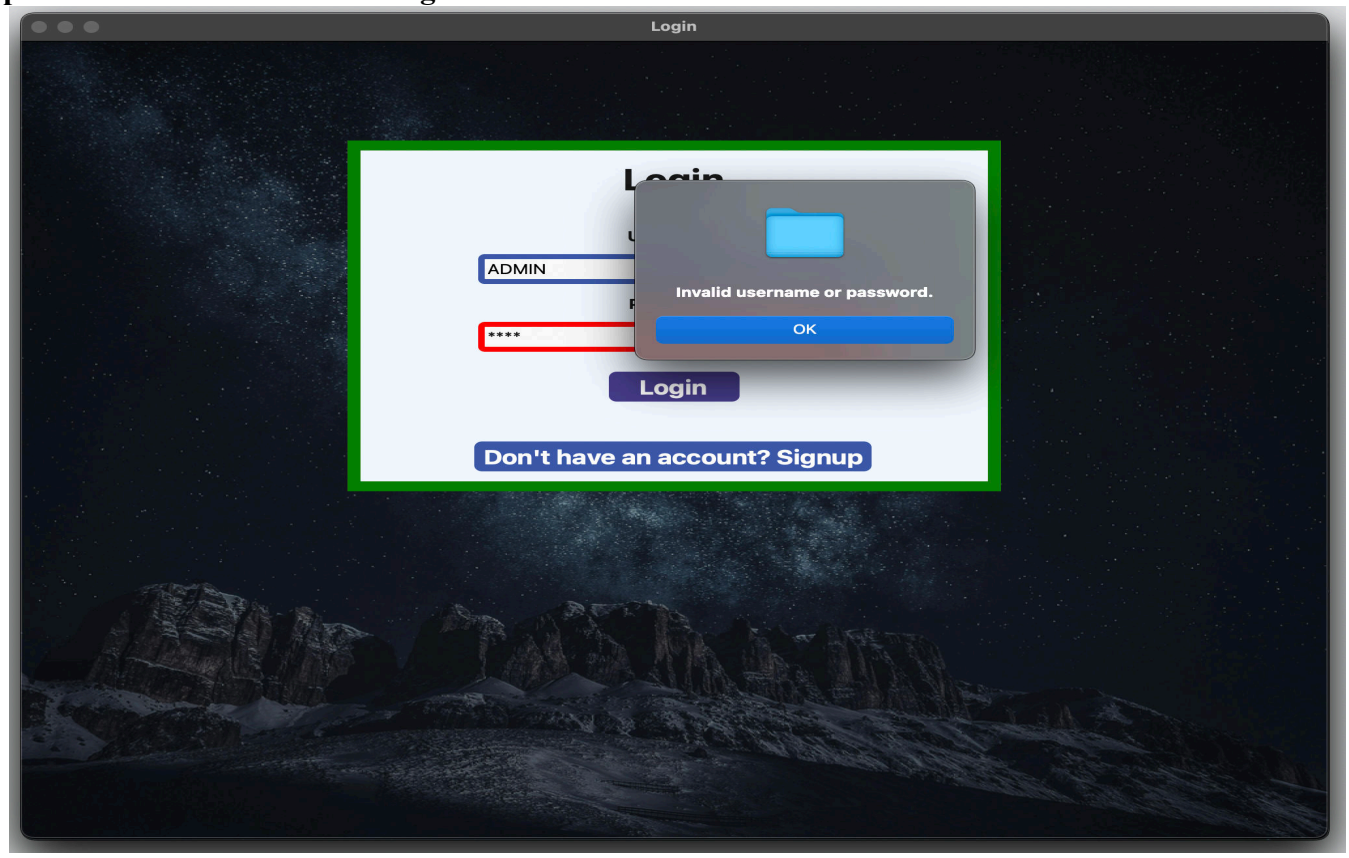


Getting Started page:

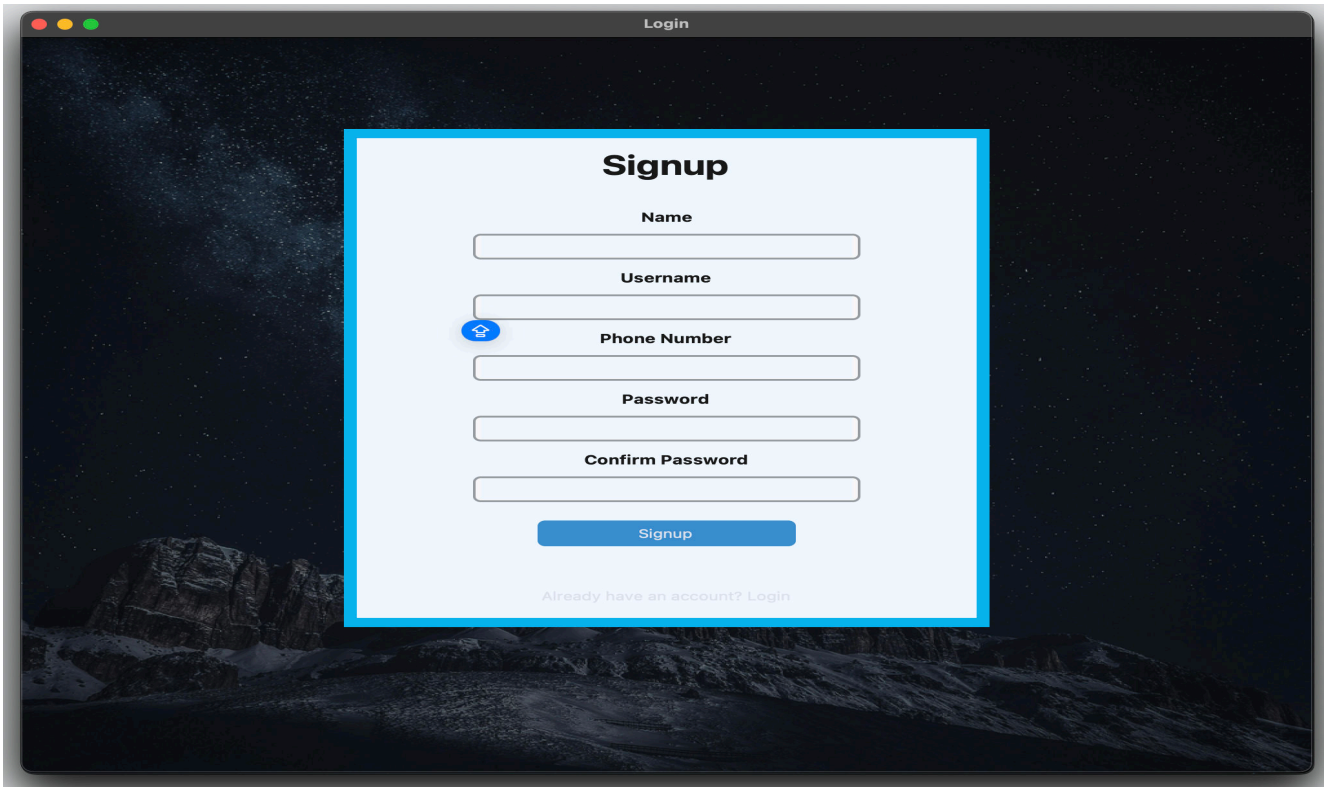
clicking “Get started” button is directed to “Login page”:



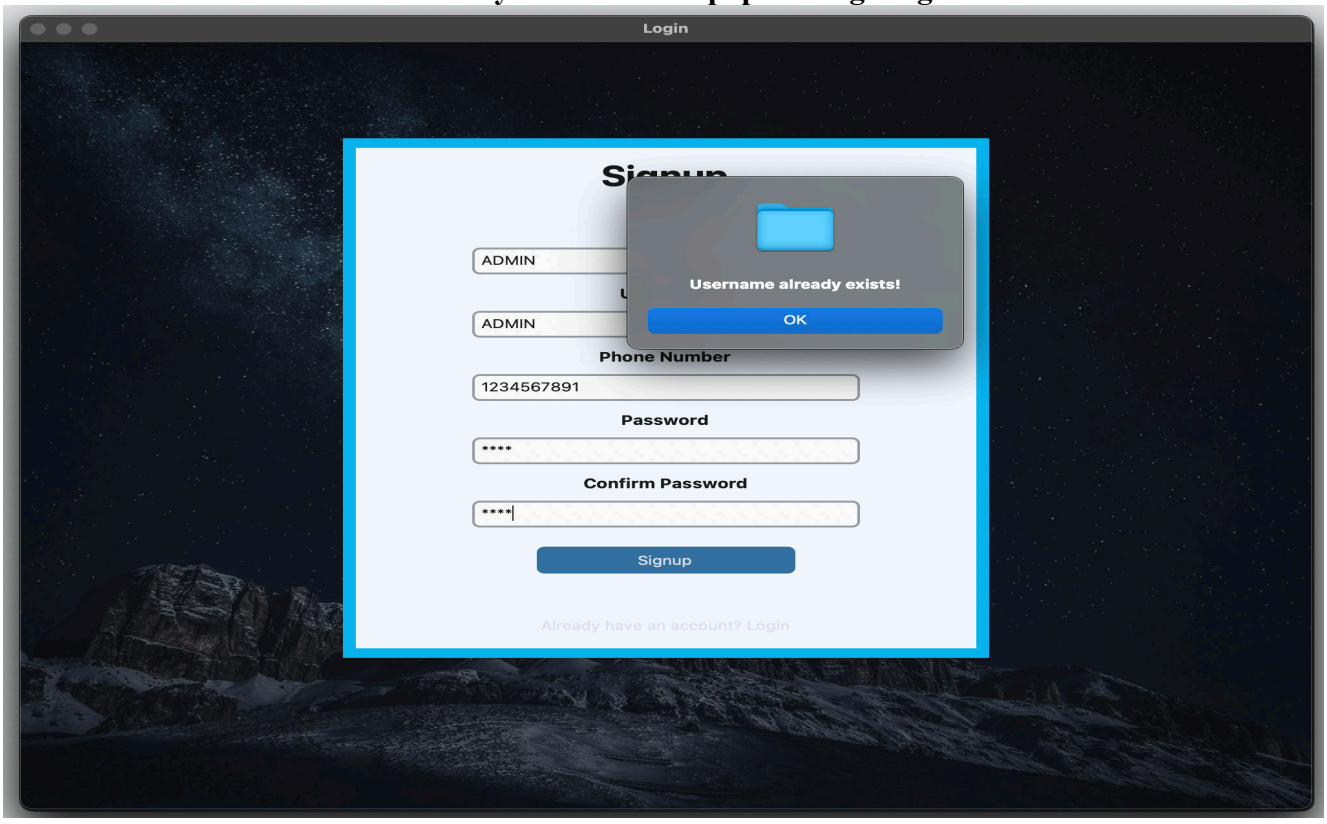
if password or username is wrong:



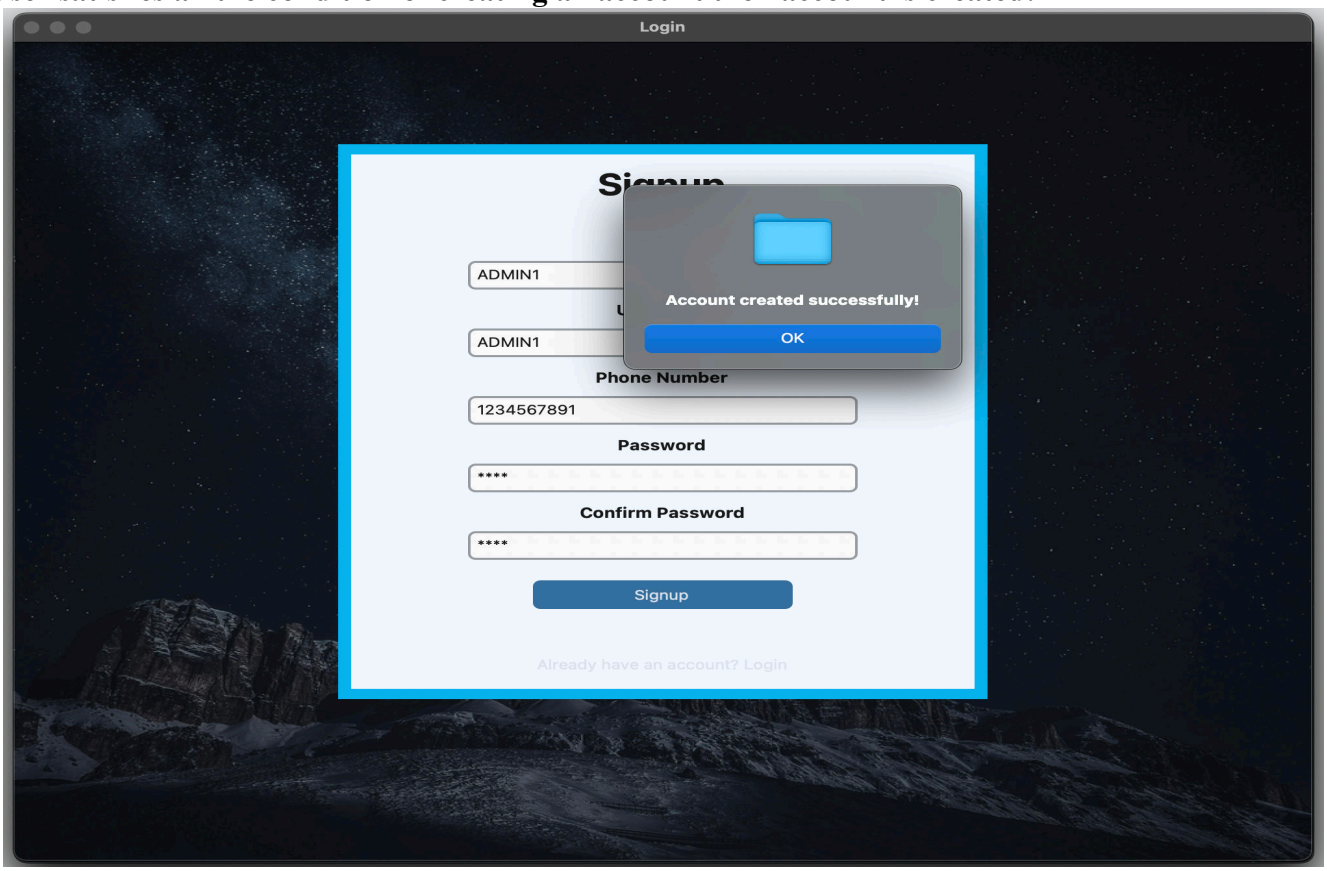
If user does not have an account then he can create an account by clicking on button “Don’t have an account? Signup”



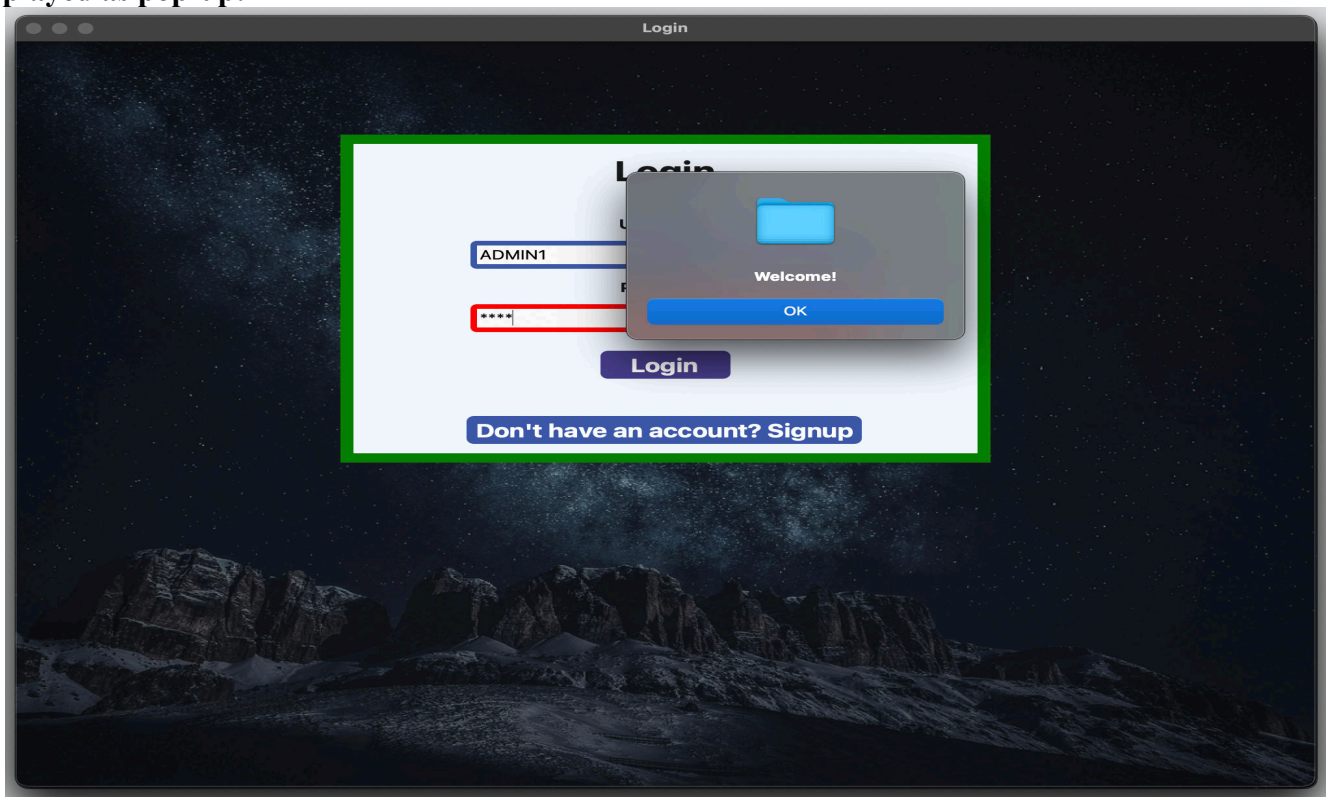
If user enters a username that is already exits then the pop message is given:



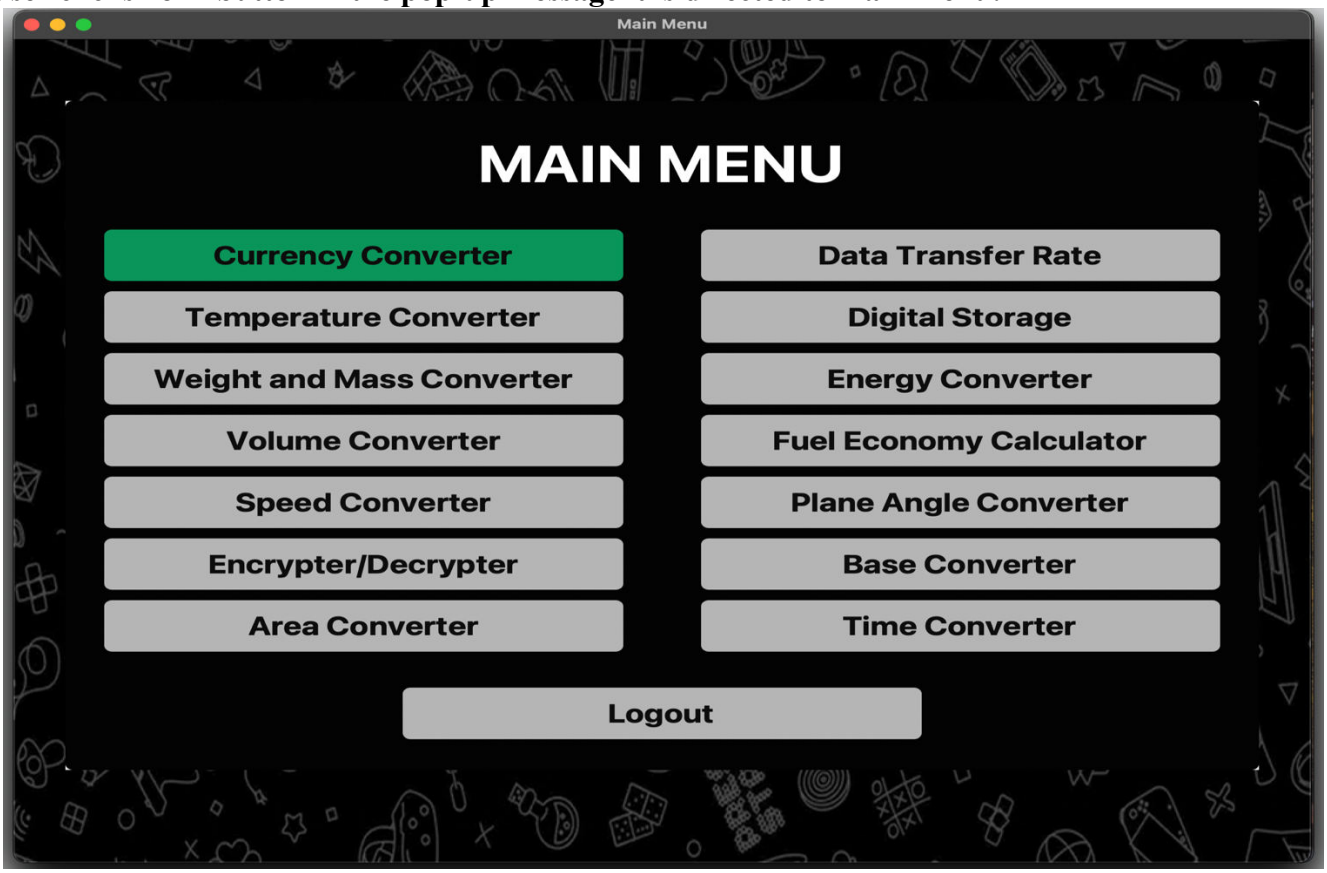
If user satisfies all the condition of creating an account then account is created:



Now the user is redirected to login page where he can enter his credentials so login welcome message is displayed as pop-up:

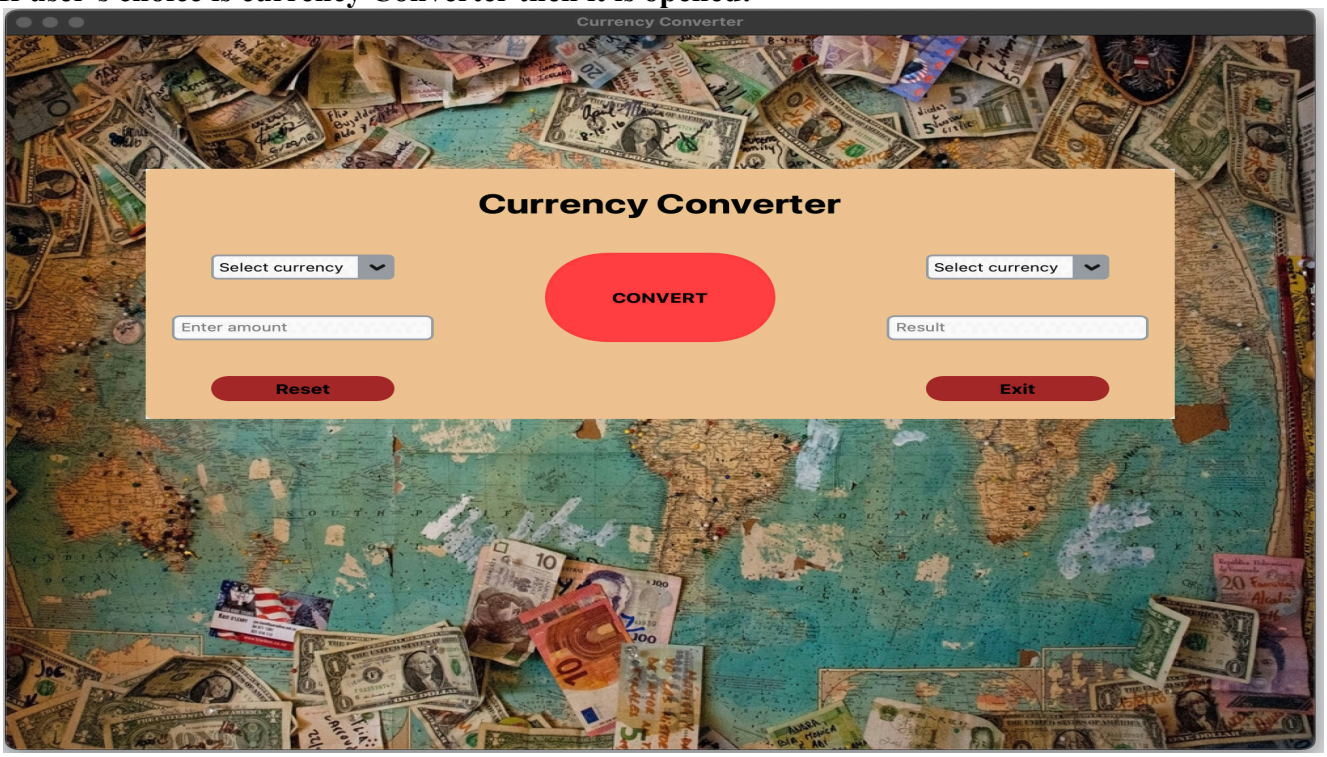


If user clicks "ok" button in the pop-up message it is directed to main menu:

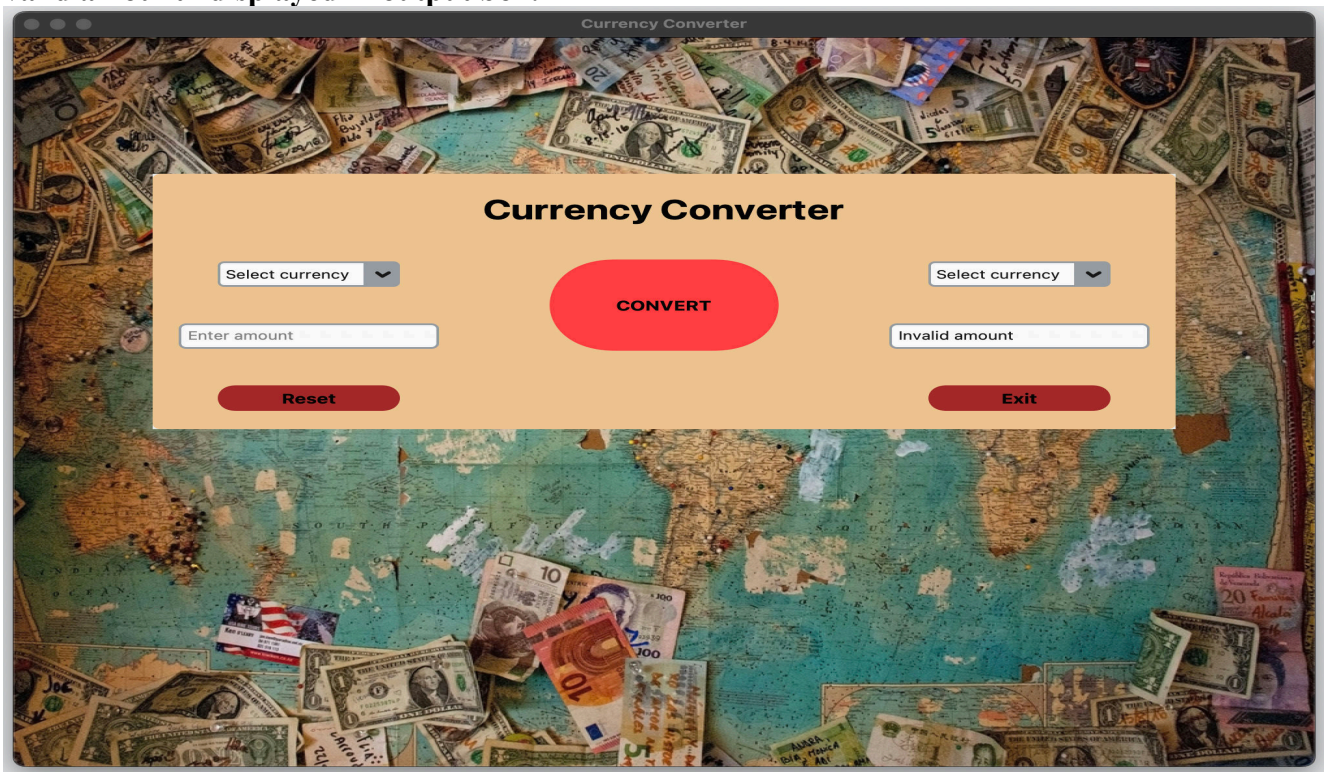


Now user can choose what conversion does he/she want as per his/her choice. Some of conversions and results are below one by one:

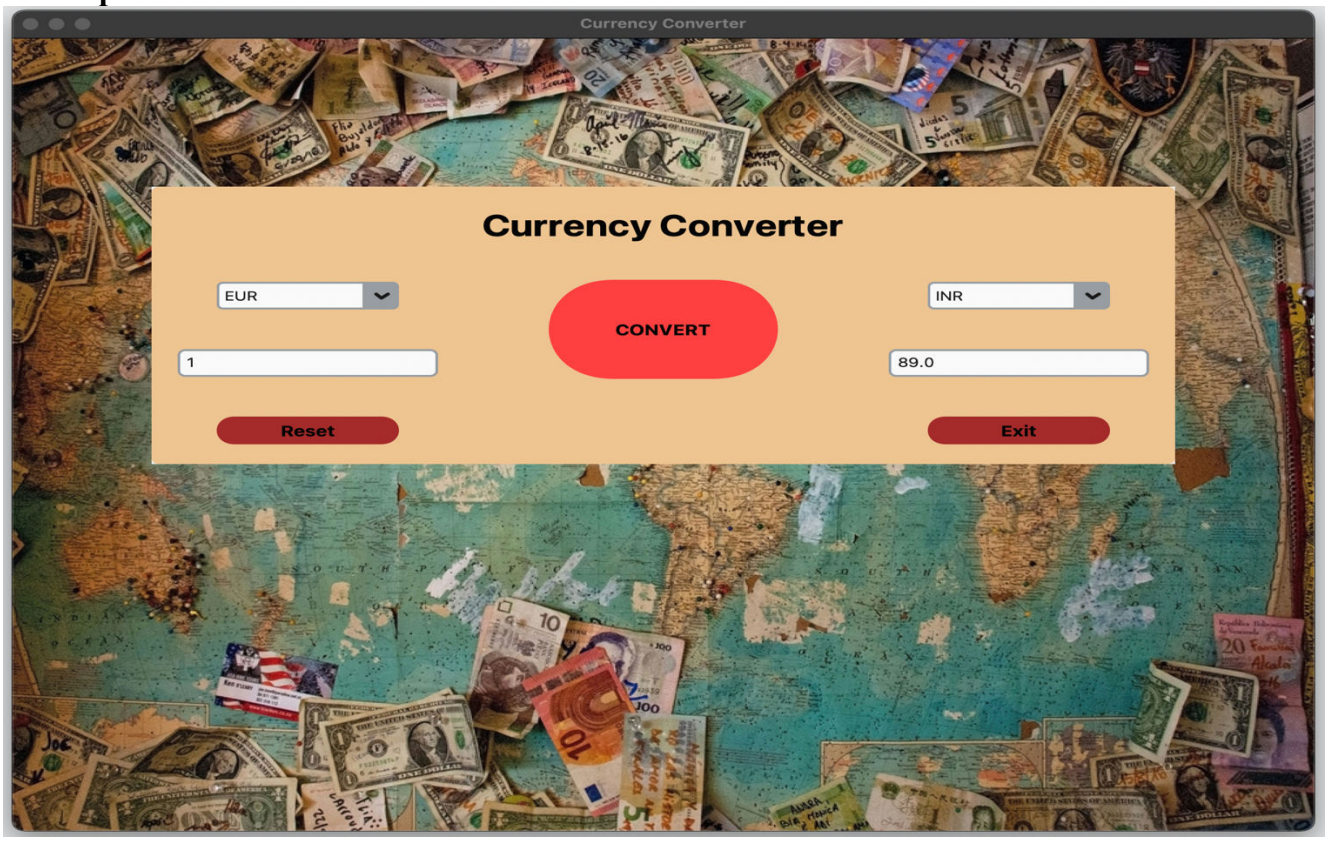
1. If user's choice is currency Converter then it is opened:



Now if user clicks by mistake on covert without typing or user input wrongly any amount value then a "invalid amount" displayed in output box:

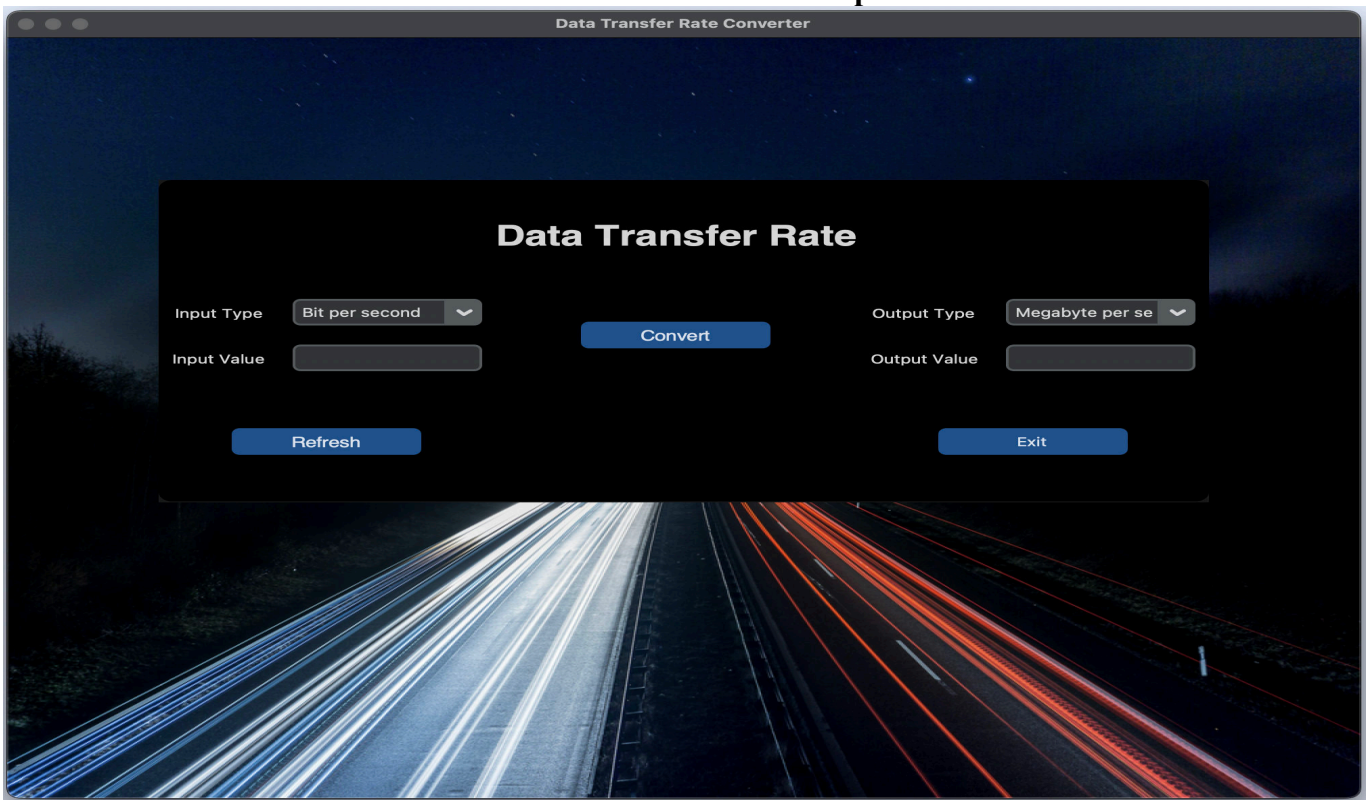


For a sample test case for conversion lets consider 1 EUR now it is converted to INR which is 89.0

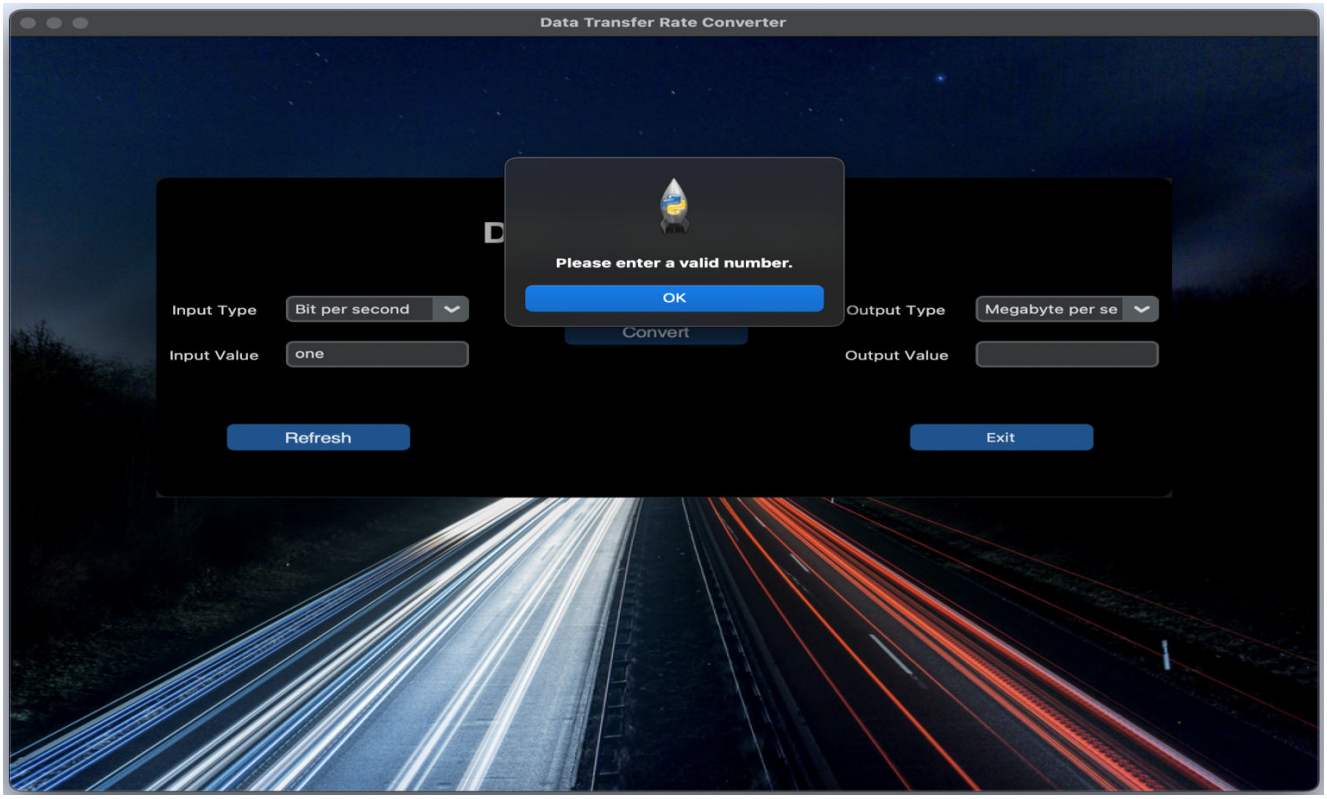


If user clicks on the exit button then it is redirected to main menu so he/she can be able to do perform multiple conversions.

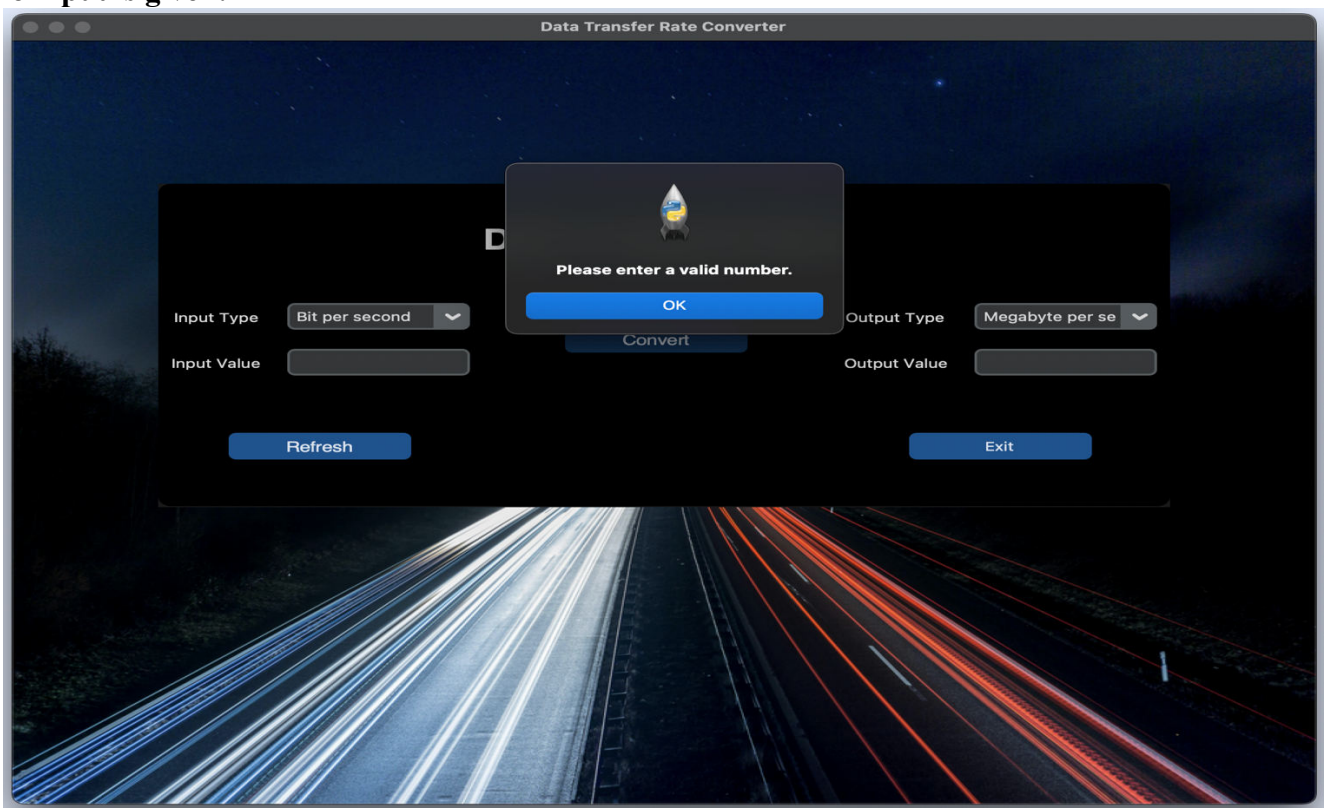
2. If user's choice is "Data Transfer Rate Converter" then it is opened:



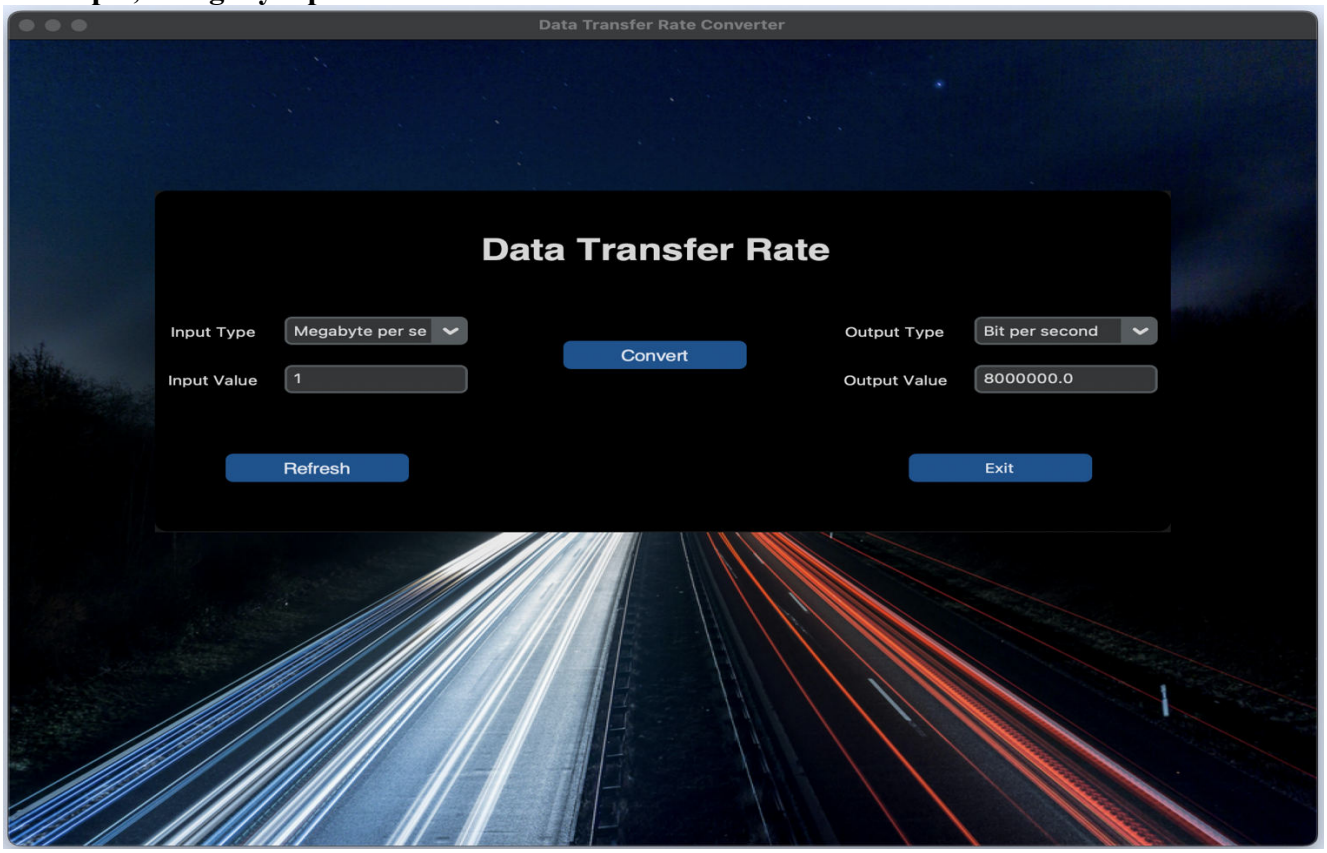
Some of failed test cases like invalid input and no input results a pop-up alert box telling invalid number:



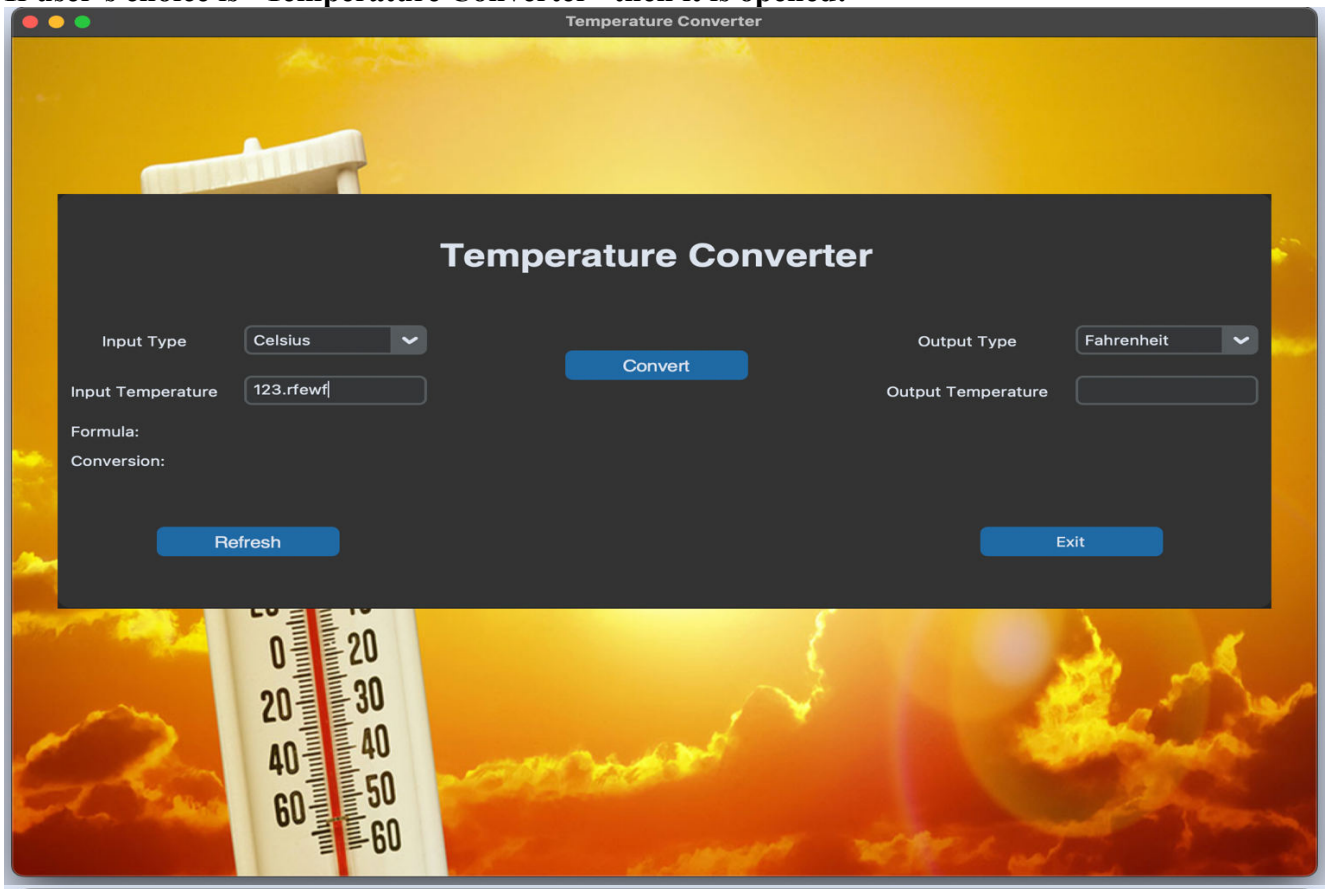
If no input is given:

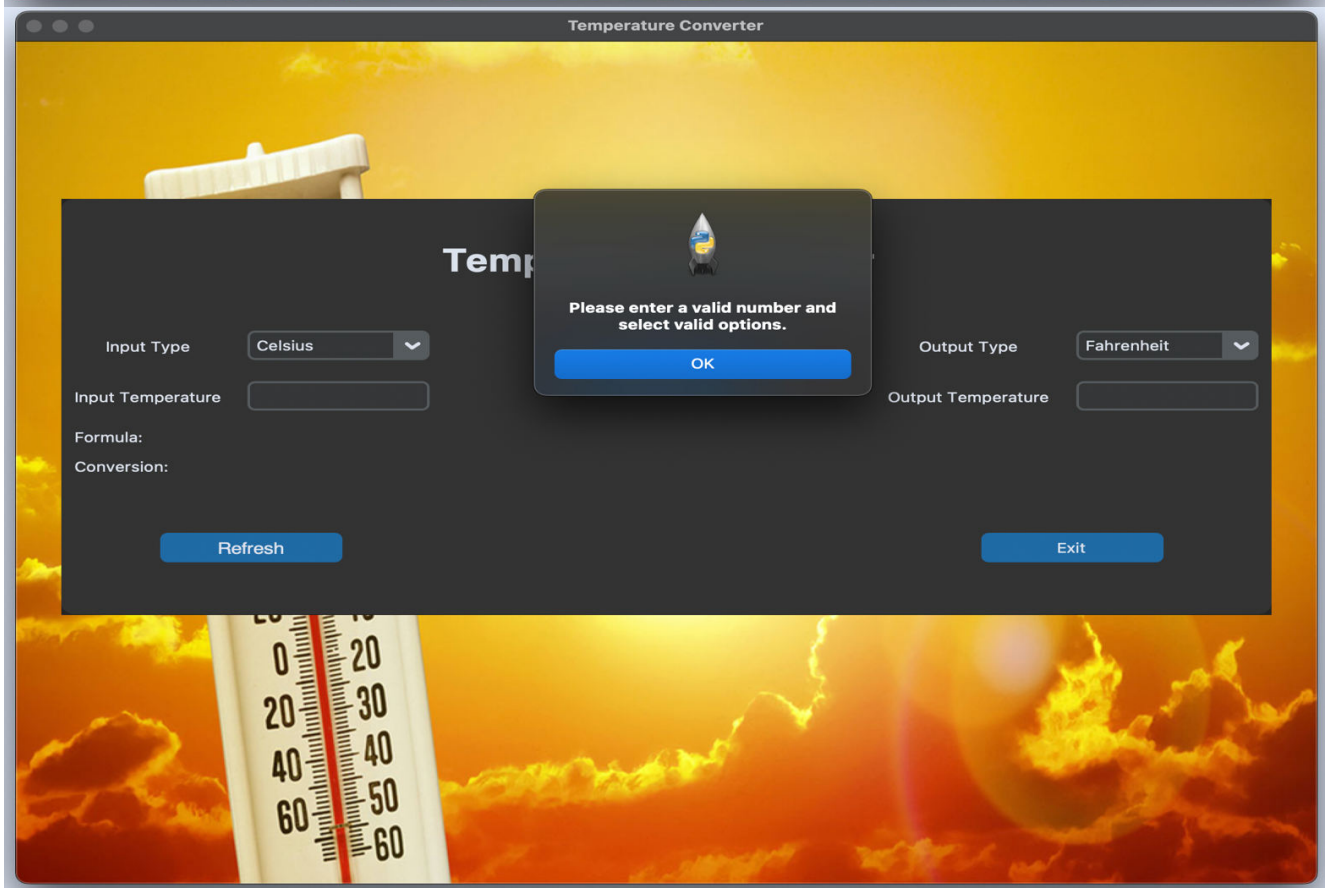


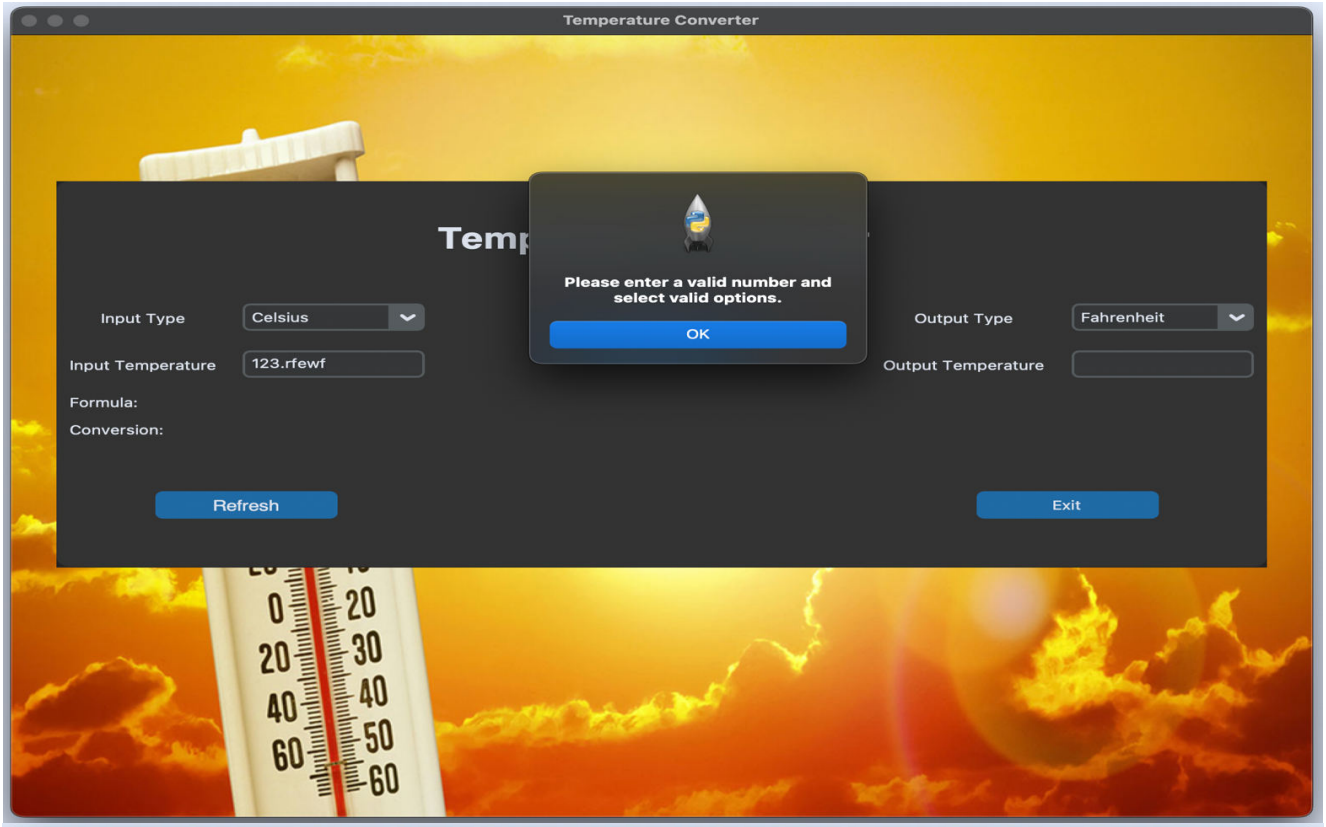
For example,1 Megabyte per Second is converted into Bit Per Second

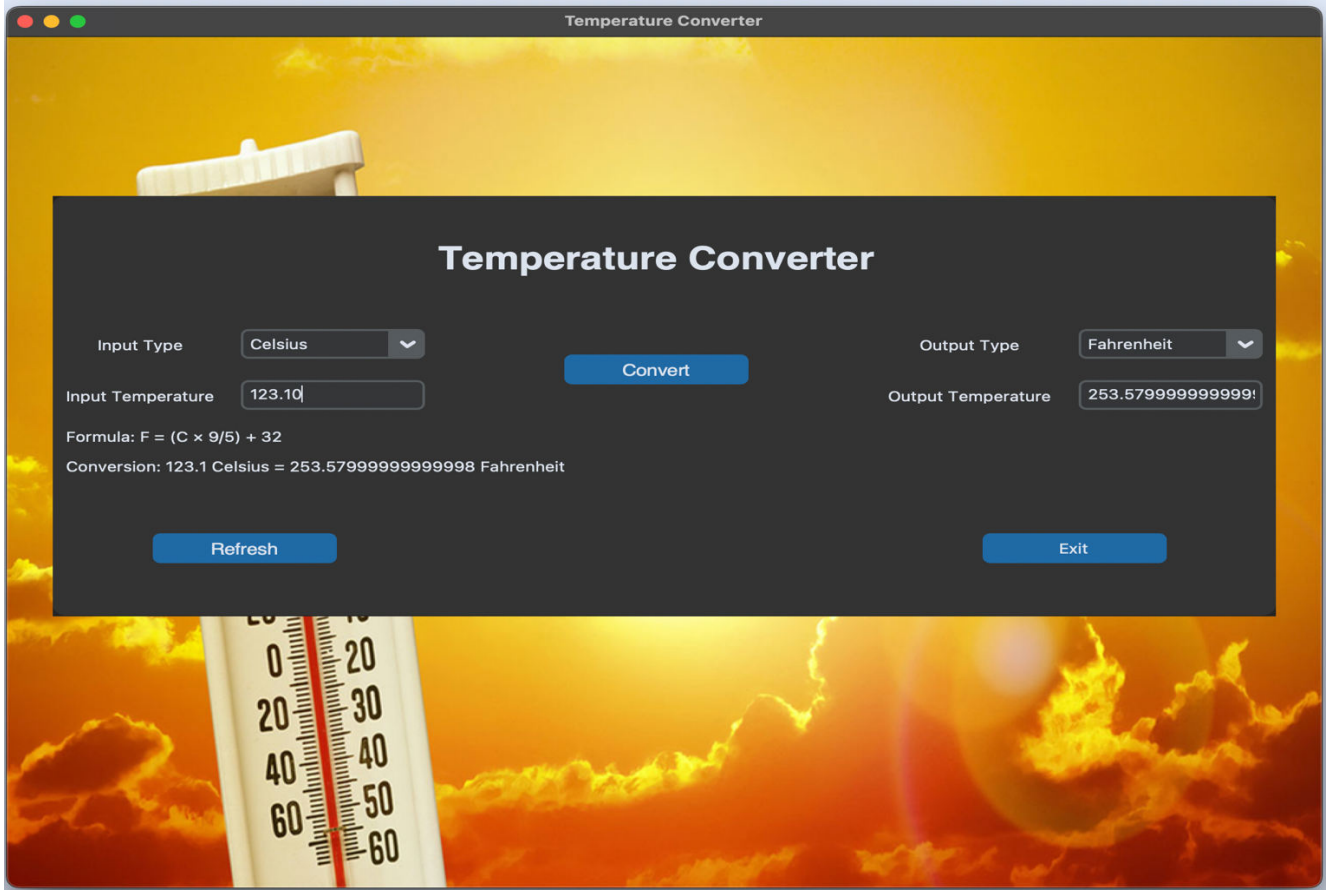


3. If user's choice is "Temperature Converter" then it is opened:

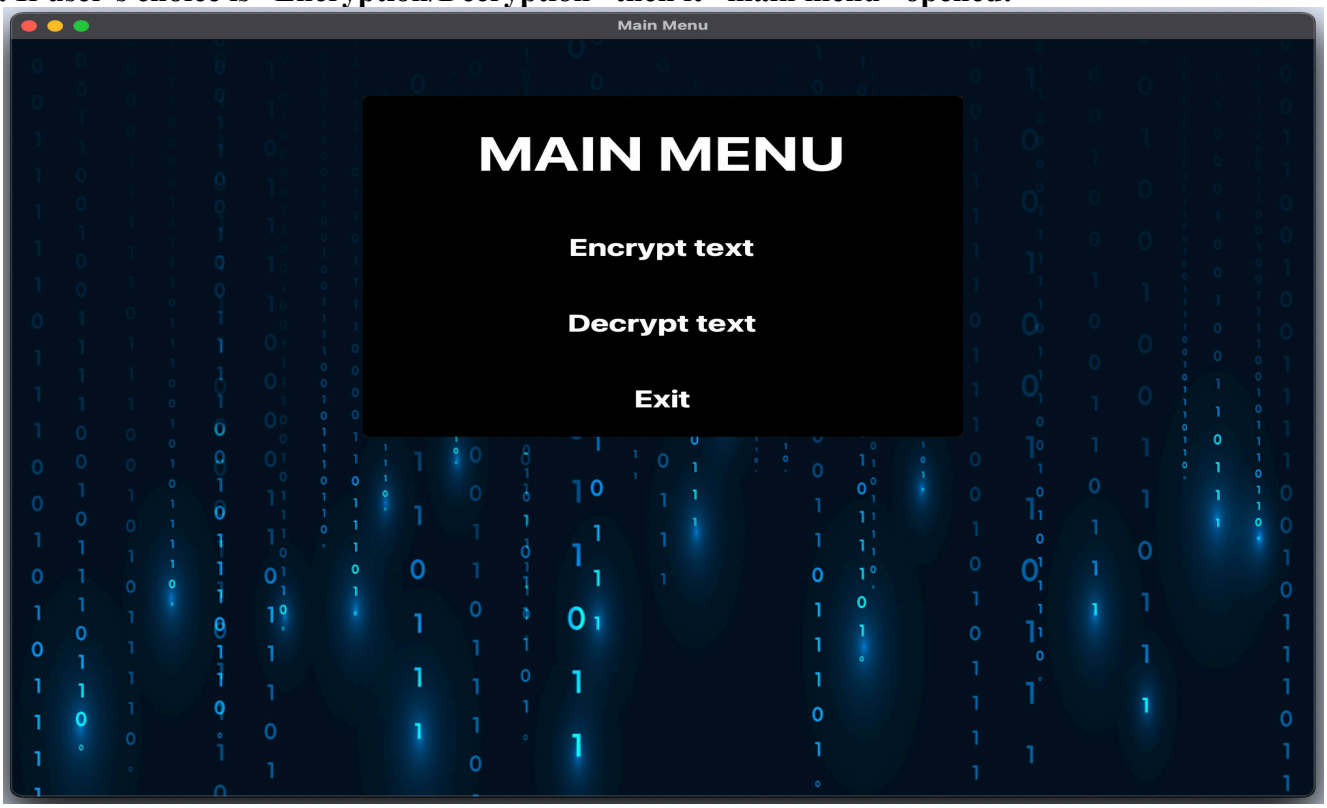




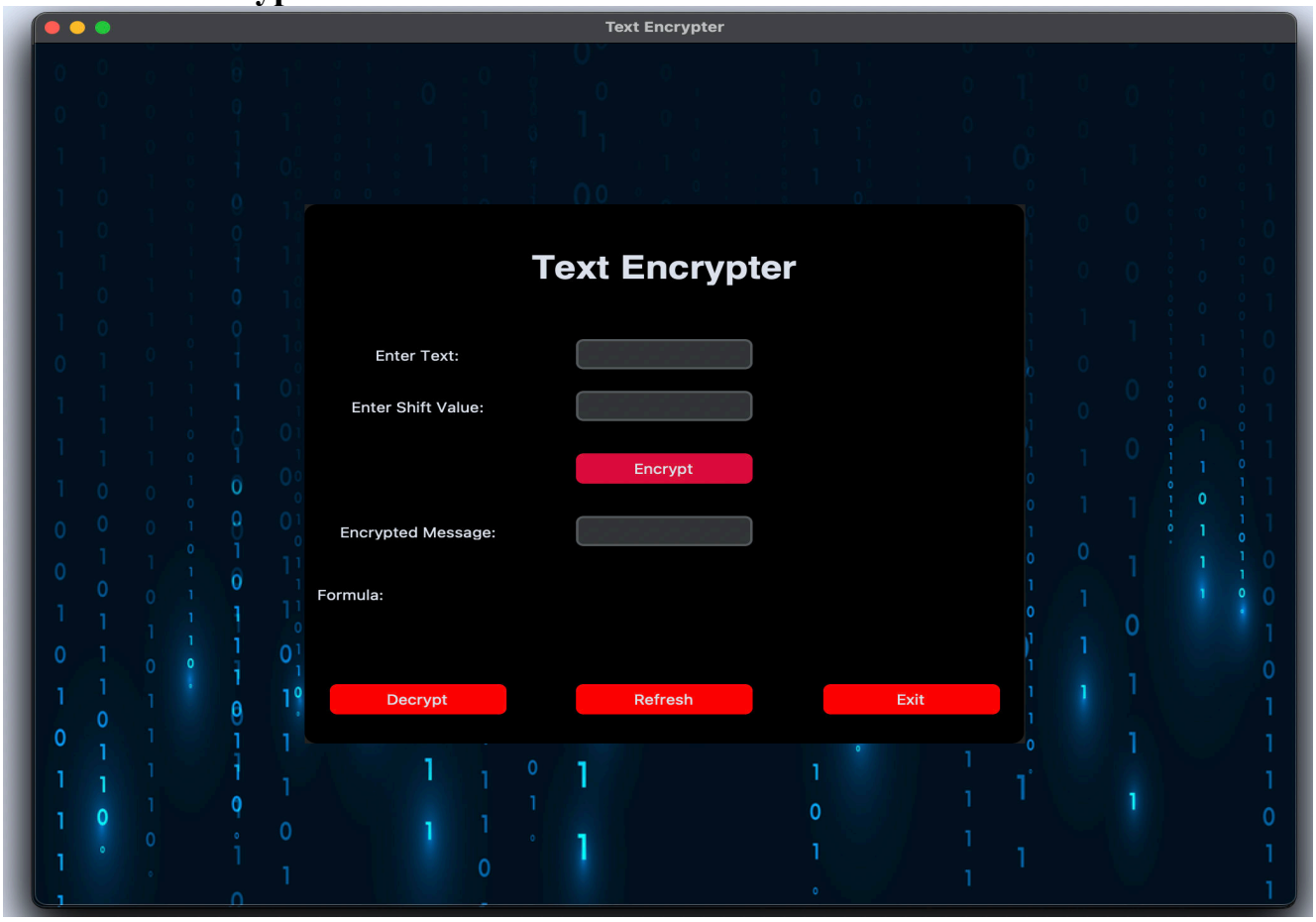

Temperature Converter
Please enter a valid number and select valid options.
OK
Input Type
Celsius
Output Type
Fahrenheit
Input Temperature
123.rfewf
Output Temperature
Formula:
Conversion:
Refresh
Exit


Temperature Converter
Temperature Converter
Input Type
Celsius
Convert
Output Type
Fahrenheit
Input Temperature
123.10
Output Temperature
253.579999999999
Formula: F = (C × 9/5) + 32
Conversion: 123.1 Celsius = 253.57999999999998 Fahrenheit
Refresh
Exit

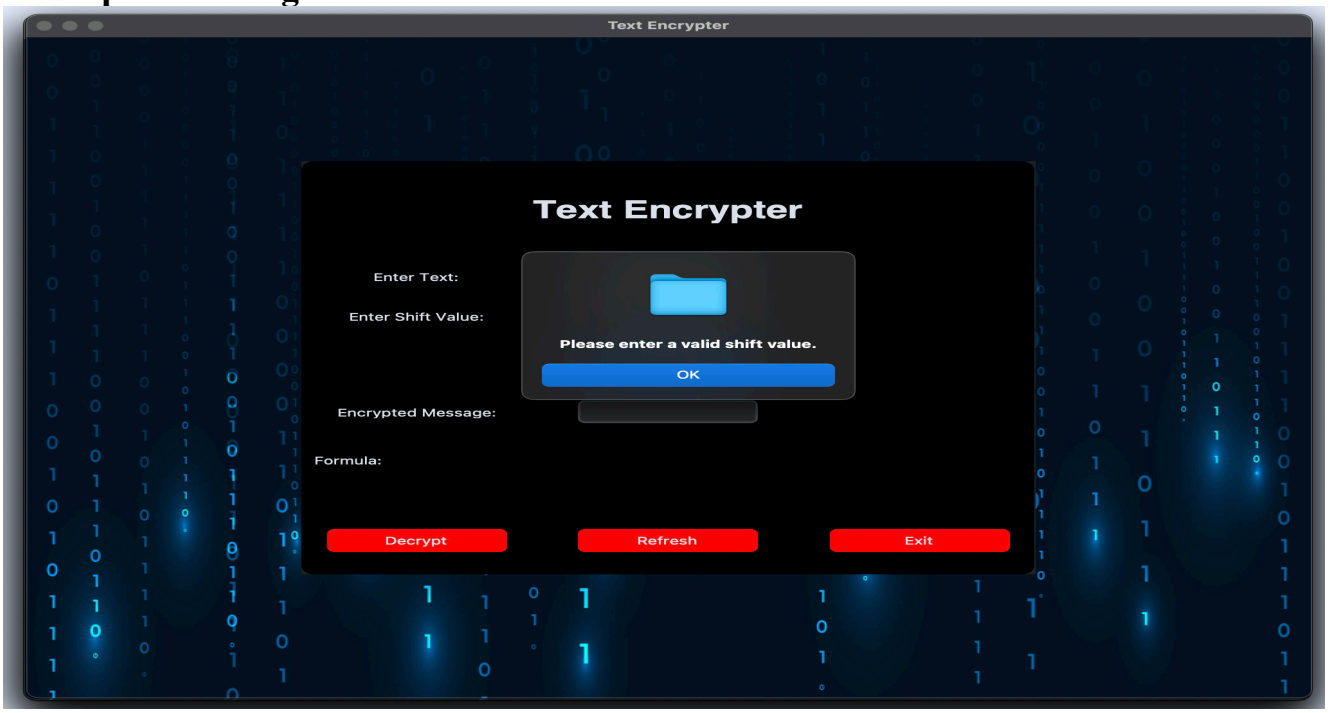
4. If user's choice is "Encryption/Decryption" then it "main menu" opened:



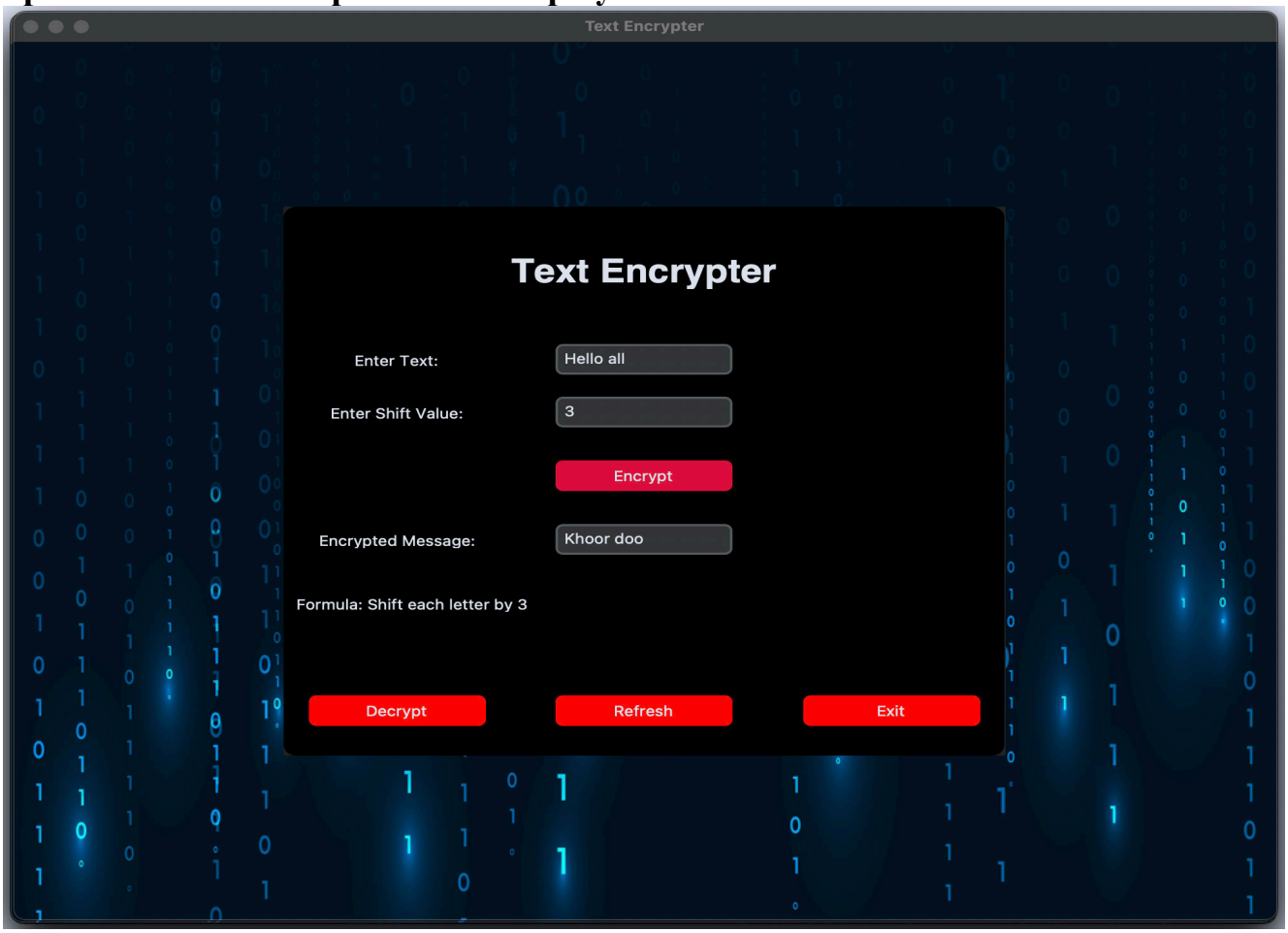
If user clicks "encrypt text" then:



If user input is wrong then:



If input is valid then output will be displayed:



7. TESTING AND VALIDATION

The **"ConvertEase"** project underwent testing to ensure its functionality, accuracy, and reliability. Different testing methods were employed at various stages of development to validate the application's performance and user experience.

Below is a detailed account of the testing process:

1.Unit Testing:

- Unit Testing focuses on validating the individual modules and components of the application to ensure that each conversion function operates as expected.
- It tests the core logic, calculations, and error-handling mechanisms of each converter independently.
- This phase ensures the integrity of functionalities such as currency conversion, encryption, temperature calculations, and others.

Test Case ID	Test Description	Input	Expected Output	Actual Output	Status
UT01	Validate login credentials	Enter valid/invalid credentials	Successful login or error message	Correct login/logout behavior	Passed
UT02	Test signup functionality	Enter valid/duplicate data	Account created or error message	Signup working as expected	Passed
UT03	Test currency conversion accuracy	Enter valid amount and currencies	Accurate conversion result displayed	Results matched external sources	Passed
UT04	Validate temperature conversion logic	Enter valid temperature and units	Converted value displayed accurately	Accurate results	Passed
UT05	Validate weight conversion logic	Enter valid weight and units	Accurate conversion value displayed	Conversion working as expected	Passed
UT06	Test encryption feature	Enter valid text to encrypt	Encrypted text displayed correctly	Encryption successful	Passed
UT07	Test decryption feature	Enter valid encrypted text	Original text retrieved	Decryption successful	Passed
UT08	Validate data transfer rate conversion	Enter valid data transfer rate and units	Accurate result displayed	Correct result	Passed
UT09	Test digital storage conversion accuracy	Enter valid storage values	Correct conversion result displayed	Accurate results	Passed
UT10	Test energy conversion logic	Enter valid energy values and units	Accurate conversion value displayed	Working as expected	Passed
UT11	Validate plane angle conversion	Enter valid angle and units	Accurate conversion value displayed	Conversion correct	Passed
UT12	Validate base conversion functionality	Enter a valid number and target base	Correct conversion result displayed	Base conversion successful	Passed

Table:7.1:Unit Testing

2.Integration Testing:

Integration Testing was conducted to ensure that different modules of the **"ConvertEase"** work cohesively when combined. This testing phase verifies the interaction between components, ensuring seamless navigation and correct data flow throughout the application.

Test Case ID	Test Description	Input	Expected Output	Actual Output	Status
IT01	Test login and navigation to menu page	Enter valid login credentials	User is redirected to the menu page	Navigation successful	Passed
IT02	Validate menu navigation to converters	Select any converter from the menu	Converter module opens without error	Navigation smooth	Passed
IT03	Test return to menu from converter	Use "Back to Menu" button	User is returned to the main menu	Transition seamless	Passed
IT04	Verify data flow between modules	Perform conversions and navigate between modules	Results and navigation are consistent	Results retained correctly	Passed
IT05	Test logout functionality from a module	Logout from any converter page	Application returns to login page	Logout works as expected	Passed
IT06	Validate encryption-decryption workflow	Encrypt a text and decrypt it immediately	Original text is retrieved successfully	Encryption-decryption seamless	Passed
IT07	Test integration of currency API	Fetch live rates and convert currency	Accurate rates are displayed and used	API integration successful	Passed
IT08	Verify system behavior during navigation	Perform multiple navigations (menu to converters and back)	Application remains stable	Stable and functional	Passed

Table:7.2:Integration Testing

3.System Testing

System Testing evaluates the application as an integrated system, covering overall functionality, user interface (UI) design, user experience (UX), and performance. It tests the interaction between modules, navigation flows, responsiveness, error handling, and application stability under various conditions. The inclusion of UI responsiveness and performance under load highlights the application's readiness for real-world use.

Test Case ID	Test Description	Input	Expected Output	Actual Output	Status
ST01	Verify login/signup flow	Enter valid/invalid credentials	Successful login/signup or error	Login/signup works	Passed
ST02	Test menu navigation	Navigate to all converters and back	Smooth transitions between screens	Navigation seamless	Passed
ST03	Test currency conversion module	Perform multiple currency conversions	Results displayed accurately	Accurate and stable	Passed

ST04	Test temperature conversion module	Perform multiple temperature conversions	Accurate conversion displayed	Results matched expected	Passed
ST05	Validate error handling for empty fields	Leave input fields blank in any module	Error message displayed	Correct errors displayed	Passed
ST06	Validate logout functionality	Click logout button	Application exits to login page	Logout as expected	Passed
ST07	Test encryption/decryption workflow	Perform encryption and decryption	Original text retrieved successfully	Encryption/decryption successful	Passed
ST08	Validate energy conversion module	Perform multiple conversions	Accurate results displayed	Conversion stable	Passed
ST09	Test volume conversion module	Perform conversions for volume units	Correct results displayed	Results matched standards	Passed
ST10	Verify fuel economy calculator	Input distance and fuel consumption	Correct fuel efficiency displayed	Works as expected	Passed
ST11	Test time converter module	Perform multiple time conversions	Accurate conversion displayed	Results matched	Passed
ST12	Test plane angle converter functionality	Perform conversions for angle units	Correct conversion results displayed	Accurate and stable	Passed
ST13	Validate menu UI responsiveness	Resize menu window	Buttons and labels adjust properly	UI elements responsive	Passed
ST14	Test dropdown menu performance	Select multiple unit options repeatedly	Dropdown works without lag or errors	Dropdown functions stable	Passed
ST15	Test “Back to Menu” button	Navigate back from modules	Smooth transition to main menu	Button works as expected	Passed
ST16	Validate system performance under load	Perform 50+ conversions in a single session	Application remains stable	No crashes or slowdowns	Passed
ST17	Validate UI/UX design consistency	Navigate through all modules	Consistent UI layout and design	UI consistent and user-friendly	Passed
ST18	Test digital storage converter module	Perform storage unit conversions	Accurate results displayed	Results matched standards	Passed
ST19	Test application startup speed	Launch application	App launches within 2-3 seconds	Startup fast and smooth	Passed

Table:7.3:Unit Testing

4. Validation of Results

Validation of Results focuses on ensuring that the outputs generated by the **"ConvertEase"** are accurate, reliable, and consistent. Each module's results were compared with trusted external sources or standard reference values to confirm their correctness.

Test Case ID	Test Description	Input	Expected Output	Actual Output	Status
VR01	Validate currency conversion accuracy	Convert USD to INR (e.g., 100 USD)	Accurate INR value displayed	Matches real-time exchange rates	Passed
VR02	Test temperature conversion correctness	Convert 100°F to °C	Accurate value (e.g., 37.78°C) displayed	Matches standard formula	Passed
VR03	Verify weight conversion accuracy	Convert 50 kg to lbs	Accurate value (e.g., 110.23 lbs)	Matches standard reference	Passed
VR04	Validate volume conversion accuracy	Convert 5 liters to gallons	Accurate value (e.g., 1.32 gallons)	Matches trusted sources	Passed
VR05	Test energy conversion results	Convert 1000 Joules to calories	Accurate value displayed	Matches calculation standard	Passed
VR06	Verify time conversion correctness	Convert 2 hours to seconds	Accurate value (e.g., 7200 seconds)	Matches expected output	Passed
VR07	Validate plane angle conversion results	Convert 90 degrees to radians	Accurate value (e.g., 1.57 radians)	Matches mathematical calculation	Passed
VR08	Test digital storage conversion accuracy	Convert 1024 KB to MB	Accurate value (e.g., 1 MB)	Matches standard reference	Passed

Table:7.4:Validation of results

8. CONCLUSION

The "**ConvertEase**" project successfully achieved its primary objective of providing a unified platform for diverse conversion needs. By integrating various conversion functionalities, such as currency, temperature, weight, and more, into a single application with a user-friendly graphical interface, the project addressed the inconvenience of relying on multiple tools.

Key Accomplishments:

1. **Comprehensive Functionality**: The application supports a wide range of conversion categories, including advanced features like encryption/decryption and fuel economy calculations.
2. **User-Friendly Design**: The intuitive GUI, designed using Tkinter and inspired by Figma mockups, ensures accessibility for users of all technical backgrounds.
3. **Accurate and Reliable Results**: All conversion modules were tested extensively, producing consistent and precise results validated against trusted sources.
4. **Scalability and Modularity**: The project's modular design allows for future enhancements and the addition of new features with minimal code refactoring.
5. **Educational Value**: The project showcases effective Python programming practices, including GUI development, modular design, and integration of third-party libraries.

Overall Impact:

The "**ConvertEase**" is a practical tool for personal, educational, and professional use, reducing the complexity of performing routine and specialized conversions. Despite its limitations, the project demonstrates the potential of Python and its libraries in building robust, multifunctional applications. Future iterations can address current limitations and expand the application's capabilities, enhancing its utility even further.

9.REFERENCES

1.API References:

- ExchangeRate-API: For fetching real-time currency exchange rates.
  https://www.exchangerate-api.com/

2.Design Tools:

- Figma Documentation: For user interface mockups and design planning.
  https://www.figma.com/

3.Official Documentation:

- Python Documentation:
  https://docs.python.org/
  (For understanding Python's standard libraries and syntax.)

- Tkinter Documentation:
  https://docs.python.org/3/library/tkinter.html
  (For GUI design and component usage in Python.)

- Tkinter Documentation:
  https://customtkinter.tomschimansky.com/documentation/widgets
  (For GUI design and component usage in Python.)

4.Converter Ideas was made from google converter:

- https://support.google.com/websearch/answer/3284611?hl=en-IN#unitconverter