

**CORE PROJECT**

**Report On**

**“Calendar Application”**

Under the Guidance Of

**Ameena Farooqui**

Report Done By

**Ishwari Gangadhar**

1. INTRODUCTION

The Calendar Application project provides a basic tool for managing and visualizing events on a calendar. The application supports features such as viewing monthly calendars, adding, viewing, and deleting events, using a command-line interface. This application demonstrates fundamental concepts in Python programming, including date and time manipulation, user input handling, and basic data management.

1.1 Overview:

* Briefly describe what a text-based adventure game is.
* Introduce the concept of the game you developed.

1.2 Purpose:

* Explain the purpose of creating the game.
* Mention the learning objectives and skills targeted (e.g., Python programming, decision-making logic).

1. OBJECTIVE

The primary objective of this project is to develop a simple, functional calendar application using Python. The application aims to:

* Display a monthly calendar for a specified year and month.
* Allow users to add, view, and delete events on specific dates.
* Provide a command-line interface for user interaction.
* Develop a simple and intuitive interface that allows users to navigate through months and years effortlessly.
* Ensure the interface is visually appealing and easy to use for people of all tech-savviness levels.

1. BACKGROUND

Calendars are essential tools for managing time and events. This project leverages Python's calendar and datetime modules to build a basic calendar application. The functionality includes viewing a monthly calendar, managing events, and interacting with users through a console-based interface.Similar to widely used applications like Google Calendar and Microsoft Outlook, the design aimed to combine simplicity with powerful functionality. The idea was to leverage Python's capabilities to develop a robust application that could be easily integrated into users' daily routines. Influenced by the sleek design and usability of modern calendar tools, the project focused on providing a seamless user experience while ensuring all essential features were included.

1. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements:

Laptop:

* Brand: Lenovo
* Model: Any model with the following specifications (adjust according to your actual laptop specifications if needed):
  + Processor: Intel Core i3 or higher (or equivalent AMD processor)
  + RAM: 4GB minimum, 8GB recommended
  + Storage: At least 10GB of free space
  + Operating System: Windows 10/11, macOS, or a Linux distribution
  + Display: Standard display resolution (1366x768) or higher

Additional Peripherals:

* Keyboard
* Mouse or touchpad
* Internet connection (for downloading software and libraries)

4.2 Software Requirements:

Operating System:

* Windows 10/11, macOS, or a Linux distribution

Python Environment:

* Anaconda Distribution (latest version)
  + Includes Python interpreter and Jupyter Notebook

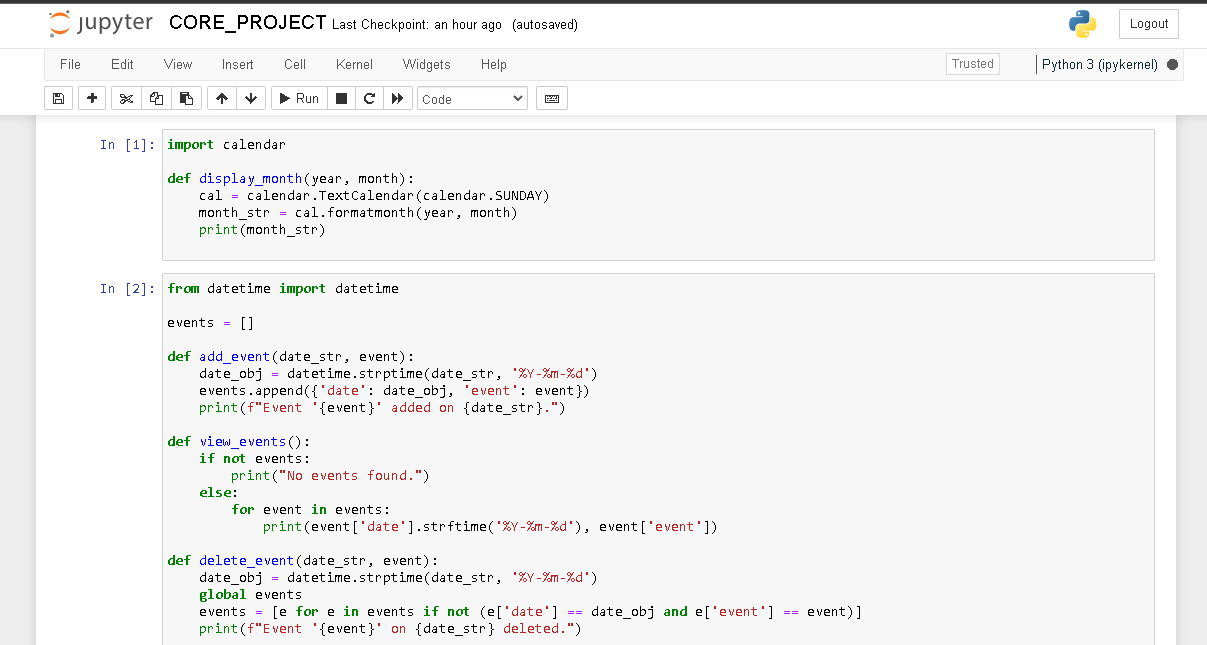
Integrated Development Environment (IDE):

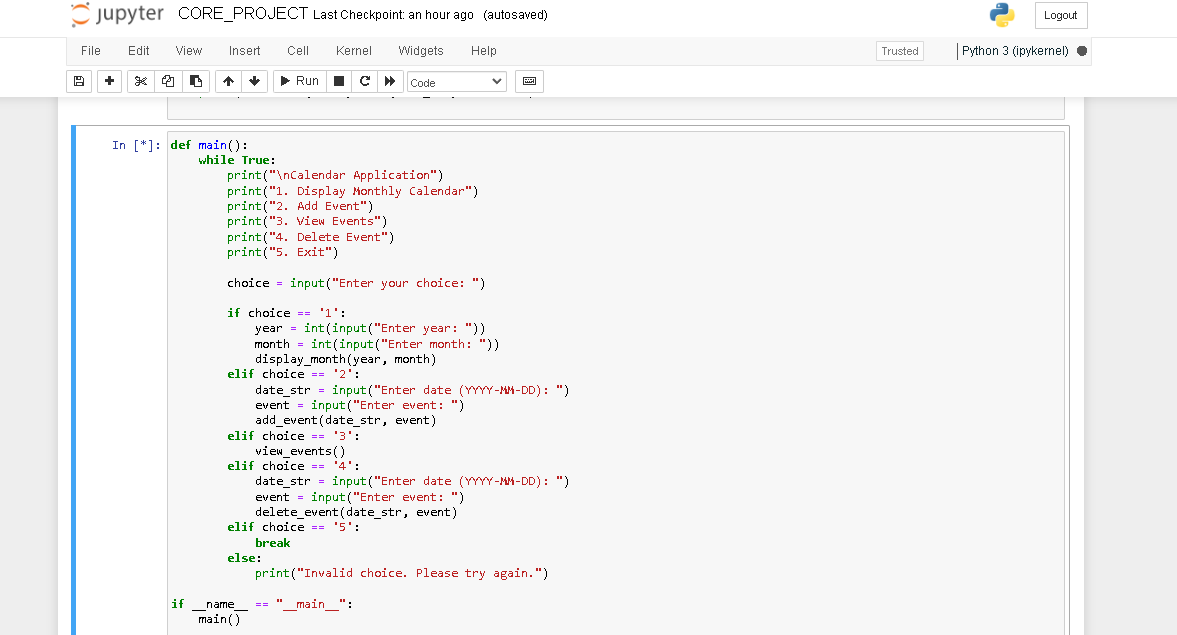
* Jupyter Notebook (included in Anaconda)

Libraries and Dependencies:

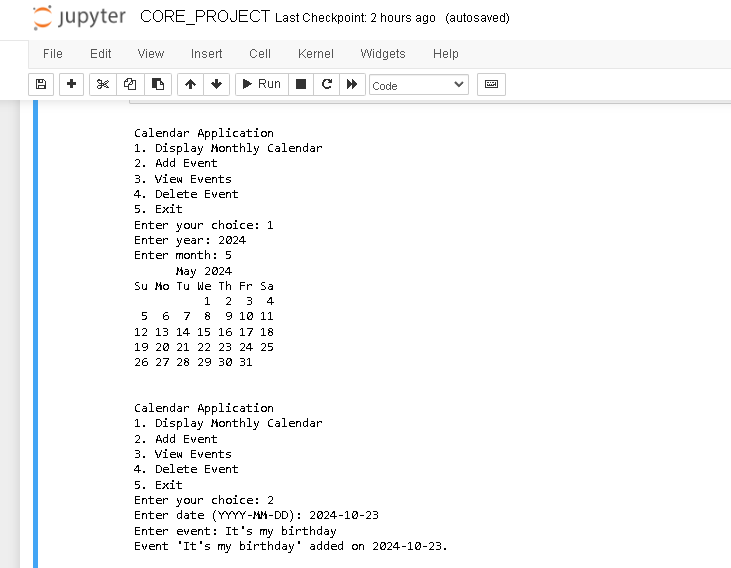
* Standard Python libraries such as sys and time
* No additional libraries are required for a simple text-based adventure game, but ensure your Anaconda environment is up to date.

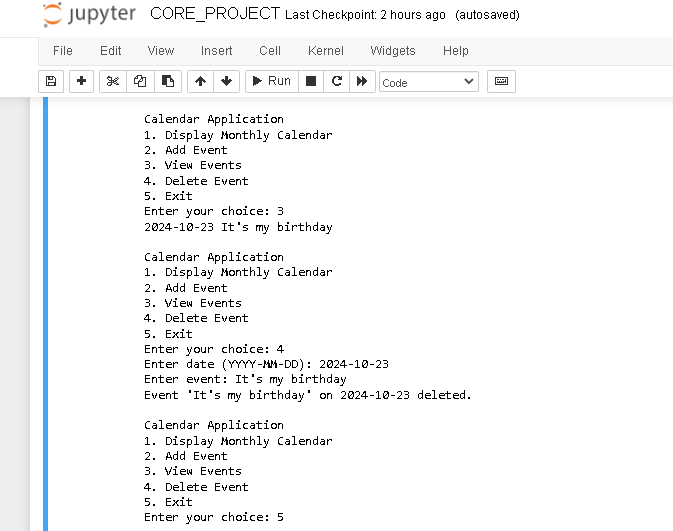
1. CODING





1. OUTPUT SCREENSHOT





1. FUTURE SCOPE

Future enhancements for this calendar application could include:

* Implementing a graphical user interface (GUI) using libraries such as Tkinter
* Adding functionality for recurring events and reminders.
* Integrating with external APIs to synchronize with online calendars.
* Adding a search feature to find events by keyword or date.

1. CONCLUSION

The Calendar Application effectively demonstrates basic calendar operations and event management using Python. It provides a useful tool for managing personal schedules and showcases fundamental programming concepts, including user interaction, date manipulation, and data storage. This report delved into the creation of a Python-based calendar application using Jupyter in Anaconda. The project aimed to build a functional tool that displays the current month, allows navigation between months, and marks specific dates. We successfully implemented these features, showcasing the efficiency of Python for creating practical applications. Key learning points include the importance of modular programming, the effective use of Python libraries, and the development of user-friendly interfaces. The project demonstrated how theoretical knowledge can be applied to real-world software development, resulting in a useful and interactive application.Personally, working on this project was enlightening and enriching. The primary challenges involved handling edge cases like leap years and ensuring an intuitive user interface. These were addressed through thorough testing and iterative design improvements based on user feedback. This project has significantly enhanced my problem-solving abilities, critical thinking, and software development skills. It has also underscored the importance of comprehensive testing and user-centric design, preparing me for future projects in the field of software engineering.

1. REFERENCES AND BIBLIOGRAPHY

* <https://docs.python.org/3/>
* <https://docs.python.org/3/library/calendar.html>
* <https://docs.python.org/3/library/datetime.html>