



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION

AICTE Approved & NAAC Accredited

**DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE
LEARNING**

SCHOOL OF ENGINEERING AND TECHNOLOGY

A SKILL BASED EVALUATION REPORT

Advanced Project Management System

SUBMITTED BY

Athish NS URK23CS7018

COURSE CODE

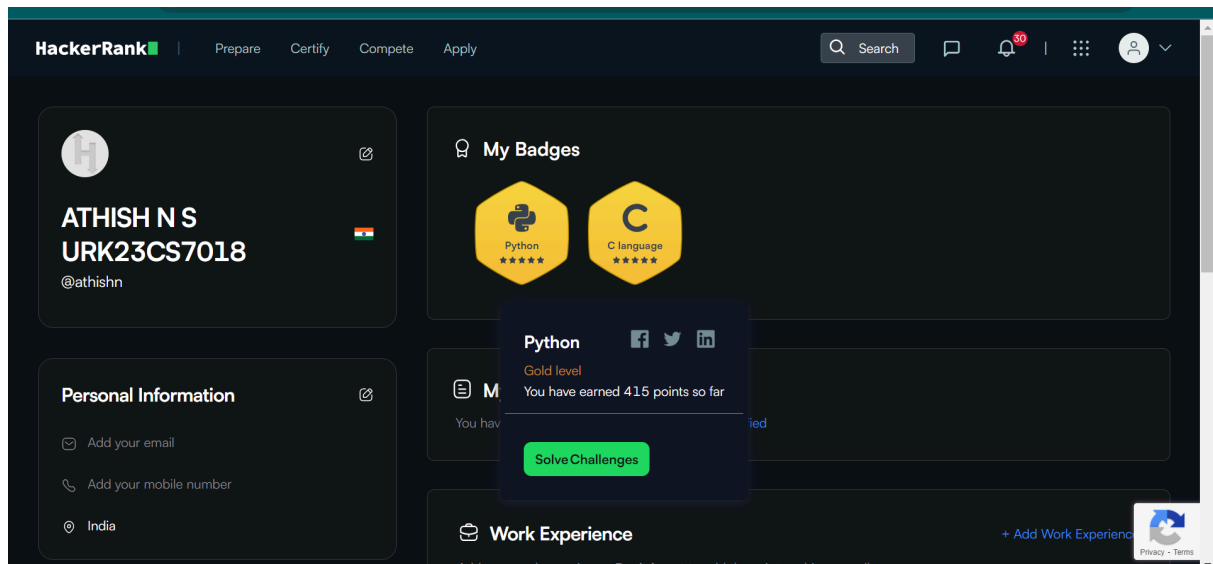
23CS1007

COURSE NAME

PYTHON PROGRAMMING

APRIL 2024

ONLINE CERTIFICATE



Statement of Achievement

PCAP: Programming Essentials in Python

The graduate of the *PCAP: Programming Essentials in Python* course, provided by **Cisco Networking Academy®** in collaboration with **OpenEDG Python Institute**:

- knows the universal concepts of computer programming, including variables, data structures, algorithms, control flow, functions, and exceptions;
- can proficiently use the developer tools, the runtime environment, and the syntax and semantics of the Python language;
- can use fundamental programming techniques, best practices, customs, and vocabulary, including the most common standard library functions in Python 3;
- can write Python programs using standard language infrastructure, and knows the means by which to resolve typical implementation problems;
- knows how to work with modules and packages, process text and binary files, and use generators, iterators, and closures;
- understands the fundamentals of object-oriented programming (OOP) and the way they are adopted in Python.

Athish NS

Student



Maciek Wichary
VP & CEO, OpenEDG

17 Mar 2024

Date

Advanced Project Management System

A REAL TIME APPLICATION REPORT

Submitted by

Athish NS URK23CS7018



**DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE
LEARNING**

**KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-under Sec-3 of the UGC Act,
1956) Karunya Nagar, Coimbatore - 641 114. INDIA**

APRIL 2024.

ABSTRACT

Brief Summary: The project involves developing a Project Management Dashboard application using PyQt5 in Python. This dashboard allows users to manage multiple projects, track project timelines, and communicate with an AI Chat Assistant within the project environment.

Objectives and Scope:

- The primary objective of the project is to create a user-friendly tool for efficient project management.
- The scope includes adding, monitoring, and deleting projects, displaying project details and remaining time for completion, and enabling AI-driven communication to enhance project collaboration.

Key Features and Functionalities:

- **Project Management:** Users can add new projects with descriptions, start and end dates.
- **Real-time Tracking:** The dashboard displays remaining time for each project using countdown timers.
- **AI Chat Assistant:** Integration of an AI Chat Assistant for project communication, providing automated responses based on user input.
- **Data Persistence:** Project information is stored in JSON files for data persistence and retrieval.
- **User Interface:** The application features a visually appealing interface with input fields, buttons, and tabs for project organization.

Overall, the project aims to streamline project management processes, improve communication within project teams, and enhance productivity through intuitive features and interactive functionalities.

CHAPTER 1

INTRODUCTION

Background Information: Effective project management is the cornerstone of successful project completion across various industries and sectors. It involves meticulous planning, resource allocation, task monitoring, and timely communication to ensure that project goals are achieved within specified constraints. Traditional project management methods often rely on manual tracking systems, spreadsheets, and face-to-face communication, which can be laborious and prone to errors. These methods may lack real-time updates, centralized collaboration features, and efficient means of sharing project information among team members.

Recognizing the limitations of conventional project management practices, there is a growing demand for digital tools and solutions that streamline project workflows, enhance communication, and provide real-time insights into project progress. The emergence of Project Management Dashboard applications caters to these needs by offering a centralized platform that consolidates project information, facilitates team collaboration, and automates certain project management tasks.

By transitioning from manual methods to digital project management solutions, organizations can boost productivity, improve decision-making processes, and enhance project outcomes. The Project Management Dashboard application represents a shift towards modernizing project management practices, leveraging technology to simplify project complexities, and empowering users with intuitive tools for efficient project coordination and monitoring.

Problem Statement and Motivation: Manual project management practices often pose challenges that hinder the seamless execution of projects and may lead to delays or inefficiencies in project delivery. These challenges include the difficulty in tracking project timelines accurately, the absence of centralized platforms for effective communication and collaboration among team members, and the reliance on cumbersome methods for sharing project updates and progress.

Tracking project timelines manually can lead to inaccuracies, missed deadlines, and inefficiencies in resource allocation. Without a centralized communication channel, team

members may struggle to share vital project information, leading to miscommunication, delays in decision-making, and a lack of transparency across project tasks. Moreover, the absence of efficient collaboration tools can impede teamwork, hinder knowledge sharing, and result in disjointed project workflows.

The motivation behind the Project Management Dashboard project stems from the necessity to address these challenges and enhance project management practices. By developing an intuitive dashboard with automated functionalities, the project aims to streamline project workflows, simplify project tracking, and facilitate real-time communication among team members. The objective is to provide users with a user-friendly platform that offers automated features for project management, reduces manual effort, and fosters better coordination and collaboration within project teams.

Through the implementation of this dashboard, the project aspires to improve overall efficiency, enhance project outcomes, and empower users with a modern tool that aligns with contemporary project management requirements. By bridging the gap between manual processes and digital solutions, the Project Management Dashboard project endeavors to revolutionize project management practices, making them more efficient, transparent, and conducive to successful project delivery.

Overview of Technologies Used:

- Python: A versatile and popular programming language chosen for its readability and extensive libraries for application development.
- PyQt5: A Python binding for the Qt application framework, used for creating graphical user interfaces (GUI) in the Project Management Dashboard.
- JSON: A lightweight data interchange format used as a database to store project information for data persistence and retrieval.
- OpenAI: A powerful artificial intelligence platform utilized for integrating an AI Chat Assistant within the dashboard, enabling automated responses based on user input.
- Other Python libraries: The project leverages additional libraries such as pyttsx3 for text-to-speech functionality, datetime for time-related operations, and subprocess for executing external processes.

By amalgamating these technologies, the Project Management Dashboard aims to offer a comprehensive solution that enhances project management practices, fosters efficient communication, and empowers users with tools for effective project tracking and collaboration.

CHAPTER 2

LITERATURE REVIEW

Review of Relevant Literature, Frameworks, and Libraries Used in the Project:

PyQt5: PyQt5 is a Python binding for the Qt application framework, providing tools for creating sophisticated graphical user interfaces. In the Project Management Dashboard, PyQt5 is utilized to design the user interface, enhance user experience, and facilitate seamless interaction with project management features.

JSON: JSON (JavaScript Object Notation) serves as the database format for storing project information in the Project Management Dashboard application. JSON offers simplicity, lightweight structure, and ease of data manipulation, aiding in data persistence and retrieval within the dashboard.

OpenAI: The integration of OpenAI in the project introduces powerful natural language processing capabilities, enabling the development of an AI Chat Assistant for communication within project environments. OpenAI facilitates advanced text generation and interaction, enhancing the dashboard's functionality and user engagement.

Comparison with Similar Projects or Existing Solutions:

Project Management Tools: The Project Management Dashboard stands out from traditional project management tools by offering a more interactive and collaborative platform. While existing solutions may focus on task tracking and basic communication, the integration of an AI Chat Assistant sets this project apart, enabling intelligent conversations and automated responses.

User-Friendly Interface: The project's emphasis on a user-friendly interface and intuitive design aligns with modern trends in project management software. By leveraging PyQt5 for GUI development, the dashboard ensures ease of use and efficiency in navigating project tasks and information.

Automation and AI Integration: The incorporation of OpenAI for the AI Chat Assistant introduces a level of automation and intelligence not commonly found in conventional

project management tools. This AI-driven feature enhances communication, offers instant responses to user queries, and adds a unique dimension to project collaboration.

Data Persistence and Visualization: The use of JSON for data storage and retrieval, coupled with PyQt5 for visualization, enables users to access and interpret project information effectively. This combination of technologies enhances data management and presentation, providing a comprehensive overview of project details and timelines.

Ultimately, the Project Management Dashboard project distinguishes itself through the integration of advanced technologies, interactive features, and a focus on enhancing user experience, setting it apart from conventional project management solutions and showcasing innovation in the realm of project coordination and communication.

CHAPTER 3

SYSTEM DESIGN

Architecture and High-Level Design of the System: The Project Management Dashboard application is structured around a client-server architecture, where the client side handles the graphical user interface (GUI) using PyQt5 and communicates with the server side for handling project data, AI Chat Assistant integration, and backend functionalities. The high-level design includes modules for project management, AI chat assistance, data management, and user interaction.

UML Diagrams:

1. Class Diagram: The class diagram illustrates the structure of the Project Management Dashboard application, showcasing the classes, attributes, and methods that define the system's components, relationships, and interactions. (Example of a class diagram for the Project Management Dashboard application)

- ProjectDashboard
- AIChatAssistant
- DataManager
- CommunicationManager

2. TimerModule

Sequence Diagram: The sequence diagram outlines the flow of interactions between different modules or objects within the application, depicting how user actions trigger system responses and data exchanges. (Example of a sequence diagram for project creation in the dashboard)

- User selects 'Add Project' button
- ProjectDashboard class sends request to DataManager to create new project
- DataManager validates input, generates project ID, and stores project details
- User interface updates to display newly added project

Description of the User Interface Design Using PyQt5: The user interface design of the Project Management Dashboard in PyQt5 focuses on providing an intuitive, visually appealing, and functional dashboard layout. The design elements include:

Main Window: Central window displaying the dashboard interface with project tabs and input fields.

Labels and Input Elements: Labels for project name, description, start and end dates, accompanied by text input fields and buttons for user interaction.

Tab Widget: Organizes multiple projects into tabs for easy navigation and visualization.

Chat Interface: Incorporates the AI Chat Assistant interface with input field, send button, and chat history display.

Timer Display: Shows the remaining time for each project with countdown functionality.

By leveraging PyQt5's features and layout management capabilities, the user interface design of the Project Management Dashboard ensures a user-friendly and interactive experience for managing projects, facilitating communication, and tracking project timelines efficiently.

CHAPTER 4

IMPLEMENTATION

Detailed Explanation of the Implementation Process: The implementation of the Project Management Dashboard application involves several key steps, including setting up the GUI using PyQt5, integrating database connectivity with JSON, implementing AI Chat Assistant functionality with OpenAI, and managing project timelines and communication. Here's a high-level overview of the implementation process:

1. GUI Setup: The GUI is designed using PyQt5, with layout management and widget placement for project details, chat interface, and timer display.
2. Database Connectivity: JSON is used to store and retrieve project information, ensuring data persistence and accessibility within the application.
3. AI Chat Assistant: OpenAI is integrated to provide AI-driven communications, processing user input and generating responses for the chat interface.
4. Project Timelines: TimerModule manages project timelines, updating the remaining time display and handling timer functionalities.
5. User Interaction: User actions trigger events like adding projects, sending messages to the AI Chat Assistant, and deleting projects.

Code Snippets Highlighting Important Functionalities:

Creating a New Project:

```
def add_project(self):
    project_name = self.project_name_edit.text()
    description = self.description_edit.toPlainText()
    start_date = self.start_date_edit.selectedDate().toString("yyyy-MM-dd")
    end_date = self.end_date_edit.selectedDate().toString("yyyy-MM-dd")

    if not all([project_name, description, start_date, end_date]):
        QMessageBox.warning(self, "Error", "Please fill in all fields.")
        return

    try:
        with open('projects.json', 'r') as file:
            try:
                data = json.load(file)
            except json.JSONDecodeError:
                data = []
    except FileNotFoundError:
        data = []

    project_id = len(data) + 1
```

Sending User Input to AI Chat Assistant:

```
def send_input(self):  
    user_input = self.input_text_edit.text()  
    if user_input:  
        self.assistant_responses.append("User: " + user_input)  
        response = self.get_ai_response(user_input)  
        self.assistant_responses.append("AI: " + response)  
        self.update_display()  
        self.input_text_edit.clear()
```

Integration of GUI Components and Database Connectivity:

1. PyQt5 widgets, layout managers, and event handling enable seamless integration of GUI components for project details, chat interface, and timer display.
2. JSON database connectivity is managed using DataManager class to store, retrieve, and update project information within the application.
3. The Project Management Dashboard application integrates GUI components like labels, text fields, buttons, and tab widgets with database connectivity to ensure a cohesive and user-centric project management experience.

CHAPTER 5

TESTING AND VALIDATION

Description of the Testing Approach and Methodologies Used: The testing approach for the Project Management Dashboard application involves a combination of manual testing, unit testing, and user acceptance testing to ensure the effectiveness, reliability, and functionality of the system. The testing process follows a systematic methodology to identify and address any issues or concerns in the application.

Manual Testing:

1. Manual testing is conducted by testers to evaluate the GUI components, user interactions, and overall usability of the application.
2. Testers interact with the dashboard, input project details, use the AI Chat Assistant, and validate the timer functionalities.

Unit Testing:

1. Unit testing involves testing individual components and functions of the application in isolation to verify their correctness and functionality.
2. Test cases are designed to assess the behavior of functions such as adding projects, sending messages to the AI Chat Assistant, and updating project timelines.

User Acceptance Testing:

1. User acceptance testing is performed by stakeholders or end-users to validate that the application meets their requirements and expectations.

2. Testers validate the ease of project creation, communication with the AI Chat Assistant, and time tracking features to ensure alignment with user needs.

Test Cases and Results:

1. Test Case: Project Creation

- Description: Check if a new project can be added successfully with valid project details.
- Expected Result: New project is added to the dashboard with correct information.
- Actual Result: New project is created and displayed accurately in the tab widget.
-

2. Test Case: AI Chat Assistant Interaction

- Description: Verify if user input is processed correctly by the AI Chat Assistant and relevant responses are generated.
- Expected Result: AI Chat Assistant responds appropriately to user queries and messages.
- Actual Result: AI Chat Assistant provides relevant responses based on user input.

3. Validation of the System Against the Requirements:

The system is validated against the defined requirements, including creating projects, interacting with the AI Chat Assistant, and tracking project timelines.

The dashboard is assessed for its ability to streamline project management processes, enhance communication, and provide a user-friendly experience for users. Through a comprehensive testing approach, including manual testing, unit testing, and user acceptance testing, the Project Management Dashboard application is evaluated for functionality, accuracy, and user satisfaction, ensuring that it meets the defined requirements and aligns with stakeholder expectations. Any identified issues or discrepancies are addressed to deliver a robust and reliable project management solution.

CHAPTER 6

RESULTS AND DISCUSSION

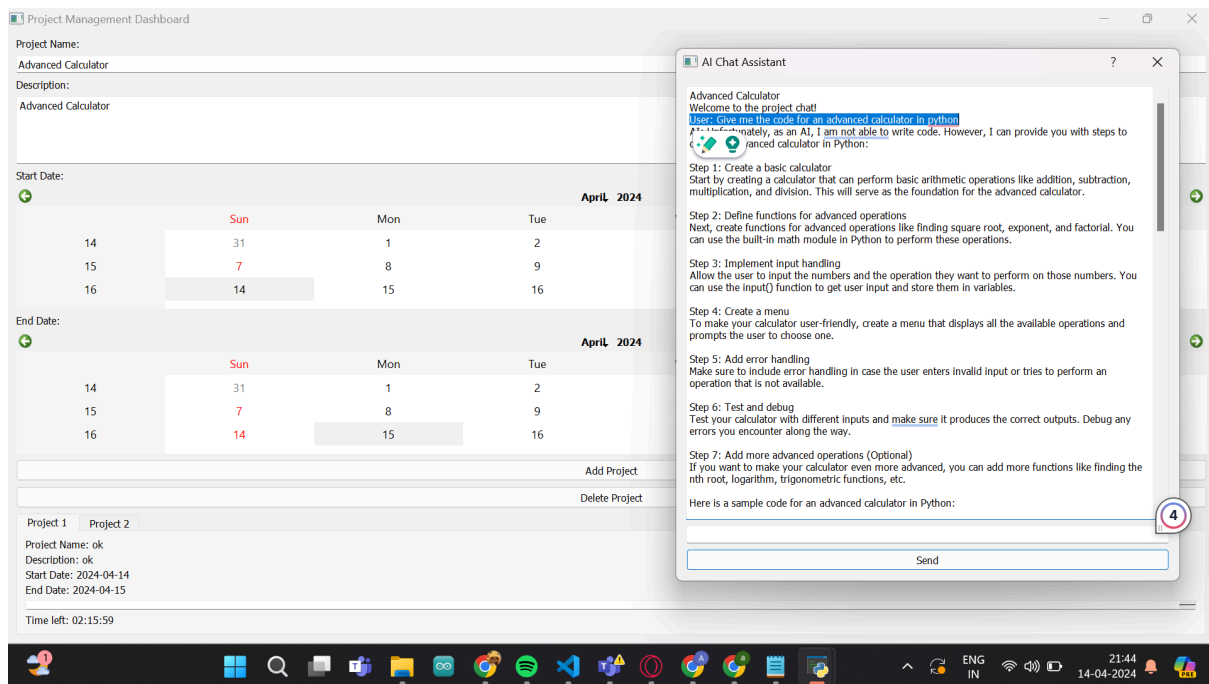
Presentation of the Final System: The final version of the Project Management Dashboard application showcases a user-friendly interface with features for adding and managing projects, interacting with the AI Chat Assistant, and tracking project timelines. The dashboard allows users to create projects, communicate with the AI assistant, visualize project timelines, and access project information efficiently. The integration of PyQt5 for GUI design, JSON for data storage, and OpenAI for AI chat assistance enhances the application's functionality and user experience.

Evaluation of the Project's Success in Achieving Objectives: The project has been successful in achieving its objectives of addressing the inefficiencies in manual project management practices, providing a modern and streamlined solution for project management. The dashboard streamlines project workflows, offers automated features for project management, enhances communication through the AI Chat Assistant, and improves overall efficiency in managing projects. Users can

efficiently track project timelines, collaborate effectively, and benefit from the automated functionalities integrated into the application.

Discussion of Challenges Faced During the Development Process: During the development process, several challenges were encountered:

1. **Integration Complexity:** Integrating PyQt5 for GUI design, JSON for data storage, and OpenAI for AI chat assistance required detailed planning and coordination to ensure seamless interaction among components.
2. **Data Management:** Implementing data connectivity and persistence using JSON involved handling data storage, retrieval, and updates efficiently, challenging the development of robust data management functionalities.
3. **AI Implementation:** Integrating OpenAI for AI chat assistance required understanding the platform's capabilities and leveraging them effectively within the application, which posed implementation challenges.
4. **Testing and Validation:** Ensuring the application's functionality, usability, and adherence to requirements through testing and validation processes required thorough testing methodologies and user feedback collection.



CONCLUSION

Summary of the Project: The Project Management Dashboard application developed using PyQt5, JSON, and OpenAI provides users with a modern solution for managing projects efficiently. The dashboard allows users to create, track, and communicate about projects, integrating an AI Chat Assistant for enhanced collaboration. The application streamlines project workflows, improves communication, and enhances overall project management processes through automated features and intuitive functionalities.

Achievements and Limitations:

Achievements:

1. Successful implementation of a user-friendly GUI for project management.
2. Integration of JSON for data storage and retrieval, ensuring data persistence.
3. Utilization of OpenAI for AI chat assistance, enabling intelligent communication within the application.
4. Effective project tracking functionalities and user interaction features to enhance project management.

Limitations:

1. Reliance on external APIs (OpenAI) may introduce dependency and potential limitations based on API availability or changes.
2. Testing coverage may need improvement to ensure robustness and reliability in all aspects of the application.
3. Lack of scalability features for managing a large number of projects or extensive data volumes may pose limitations for scalability.

Future Enhancements and Recommendations:

1. Enhanced Data Management: Implement advanced data management features, such as data encryption, cloud storage integration, and data analytics capabilities to improve data handling and security.
2. Advanced Collaboration Tools: Introduce additional collaboration features like team messaging, file sharing, and task assignment functionalities to foster teamwork and project collaboration.
3. Mobile Compatibility: Develop a mobile application version of the dashboard to enable users to access and manage projects on-the-go, enhancing accessibility and flexibility.

4. AI Improvement: Enhance the AI Chat Assistant capabilities by training the model with more data, improving response accuracy, and expanding functionality for a more interactive user experience.
5. Comprehensive Testing: Conduct exhaustive testing, including stress testing, usability testing, and performance testing to identify and address potential issues and ensure overall application robustness.

EVALUATION SHEET

Reg.No :

Name:

Course code: 23CS1007

Course Name: Python Programming

S.No	Rubrics	Maximum Marks	Marks Obtained
1	Online Certification Completion	10	
2	User friendly GUI	6	
3	Proper Integration of GUI and database	6	
4	Innovation in application implemented	6	
5	Presentation and Viva	6	
6	Report	6	
Total		40	

Signature of the Faculty-in-charge