University of Engineering & Technology, Lahore
Mechatronics & Control Engineering Department

Course Name:	Computer Programming II (MCT243L)
Title:	Problem-Based Learning (PBL)
Session:	2023
Dated:	8th December 2024
Course Instructor:	Maryam Saleem

The key idea of this project is that students will embark on a **Capstone Project Series: Building Real-World Applications with Python**, where they will develop a fully functional application that addresses a real-world problem. Each project integrates core concepts such as AI, data analysis, automation, and GUI development. Every group can choose from diverse projects like AI-powered facial recognition security systems, sentiment detectors, or vehicle detection using OpenCV, emphasizing machine learning and computer vision. Alternatively, you may opt for projects focused on data interaction, such as weather forecasting dashboards, stock market prediction apps, or language translation tools, which will highlight API integration and data visualization. GUI-centric projects like restaurant management systems, typing test games, or cafe management tools will emphasize creating user-friendly interfaces using Tkinter or PyQt, while automation-focused projects, including email senders, currency converters, and URL shorteners, will demonstrate task automation through intuitive interfaces. The goal is to design a working prototype with a clean GUI, implement core functionalities, ensure robust error handling, and document the project's development process through a brief report detailing the logic, challenges, and outcomes.

The Project Details:

The list of projects and their required libraries are given in Table 1.

Table 1: Projects list with their required libraries

Sr. no.	Project Titles	Required Libraries
1	AI-Based Real-Time Facial Recognition Security System	**opencv-python:** For video capture and image processing **face_recognition:** For facial recognition algorithms **dlib:** Dependency for face_recognition **numpy:** For array manipulation and numerical operations
2	Weather Forecasting Dashboard	**requests:** For fetching weather data via APIs **tkinter:** For building the GUI (built-in) **matplotlib:** For displaying graphical data
3	AI-Powered Chatbot for Customer Support	**nltk:** For natural language processing **tkinter:** For creating the chatbot GUI
4	Mental Health Companion App	**nltk:** For sentiment analysis **tkinter:** For GUI development **requests:** For any API-based interaction
5	Restaurant Management System GUI Application	**tkinter:** For building the GUI (built-in) **pandas:** For managing tabular data

6	Google Translator GUI Application	**googletrans:** For language translation **tkinter:** For GUI development
7	Vehicle detection in a Video frame using OpenCV	**opencv-python:** For image and video frame processing **numpy:** For handling image arrays
8	Create a GUI to extract Lyrics from song	**requests:** For fetching lyrics from APIs **beautifulsoup4:** For web scraping **tkinter:** For GUI
9	Count Objects in an Image	**opencv-python:** For image processing **numpy:** For handling image arrays
10	Sentiment Detector GUI using Tkinter	**nltk:** For sentiment analysis **tkinter:** For creating the chatbot GUI
11	Image Viewer App in Python using Tkinter	**PIL (Pillow):** For image handling and display **tkinter:** For GUI development
12	Build a stock market prediction app	**yfinance:** For fetching stock data **pandas:** For data manipulation **scikit-learn:** For machine learning models **matplotlib:** For data visualization
13	Signature-recognition System	**opencv-python:** For image processing **scikit-learn:** For machine learning model **numpy:** For data manipulation
14	Robot Builder Application by GUI	**tkinter:** For GUI development **PIL (Pillow):** For handling image components
15	Dictionary GUI Application	**tkinter:** For GUI development **json:** For storing dictionary data (built-in)
16	Email Sender GUI Application	**smtplib:** For sending emails (built-in) **tkinter:** For GUI development **email.mime:** For constructing email content (built-in)
17	Instagram User detail GUI Application	**requests:** For API interaction. **tkinter:** For GUI development
18	Convert Text to Numerical data	**sklearn.preprocessing:** For text-to-numeric conversion **pandas:** For data manipulation
19	Cafe Management System GUI Application	**tkinter:** For GUI development **sqlite3:** For database management (built-in)
20	Currency Converter GUI Application	**requests:** For fetching currency rates from APIs **tkinter:** For GUI development
21	Url Shortener GUI Application	**pyshorteners:** For URL shortening **tkinter:** For GUI development
22	Maze Solver	**networkx:** For graph-based algorithms **matplotlib:** For visualization
23	Typing Test Game using GUI	**tkinter:** For GUI development **time:** For measuring typing speed (built-in)

Libraries:

The required libraries for this project are given in detail.

- **OpenCV:**

For the OpenCV library, the package is installed using the package manager PIP. To install any library in Python, first, you must ensure a couple of path settings and then use the PIP manager. The documentation of OpenCV can be found at [1].

```
import cv2
```

In the provided code, the OpenCV library (**cv2**) must be used to read and preprocess images before training the model for image classification. Here is the code:

- **Numpy:**

The numpy library (**np**) is used primarily for handling arrays and numerical operations, especially in the context of image data. Numpy arrays can be used to efficiently handle image data, perform mathematical operations, and manipulate arrays. It converts lists or other data structures to numpy arrays for compatibility with machine learning algorithms and efficient numerical computations. The library can be installed from [2]

```
Import numpy as np
```

- **Scikit-learn:**

The scikit-learn library (**sklearn**) is used for several functionalities, including splitting the dataset into training and testing sets, training a random forest model, and evaluating the model's performance. The library can be found at [3].

- **Face_Recognition:**

Face_Recognition is a library built on dlib's state-of-the-art face recognition capabilities. It simplifies the implementation of facial recognition for detecting and identifying faces in images and video streams.

```
pip install face_recognition
```

- **Dlib:**

Dlib is a modern C++ toolkit that contains machine learning algorithms and tools for creating complex software. It's often used for facial recognition and object detection in Python.

```
pip install dlib
```

- **Requests:**

Requests is a simple yet powerful library for making HTTP requests. It's commonly used to interact with APIs, retrieve web data, or send data to web services.

```
pip install requests
```

- **Matplotlib:**

Matplotlib is a plotting library used for creating static, interactive, and animated visualizations in Python. It is commonly used for data visualization in machine learning and scientific computing.

```
pip install matplotlib
```

- **NLTK (Natural Language Toolkit):**

NLTK is a powerful library for working with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources for natural language processing tasks.

```
pip install nltk
```

- **Pandas:**

Pandas is a data manipulation and analysis library that provides powerful data structures like DataFrames for managing structured data. It is widely used for cleaning, analyzing, and transforming datasets.

```
pip install pandas
```

- **Pillow (PIL):**

Pillow is a Python Imaging Library (PIL) fork that adds support for opening, manipulating, and saving image files in various formats such as PNG, JPEG, and BMP.

```
pip install pillow
```

- **Yfinance:**

Yfinance is a library that provides a simple interface to access historical market data from Yahoo Finance, including stock prices and financial metrics.

```
pip install yfinance
```

- **Googletrans:**

Googletrans is a free and unlimited Python library that uses Google Translate's API to provide translations between multiple languages.

```
pip install googletrans==4.0.0-rc1
```

- **BeautifulSoup4:**

BeautifulSoup4 is a web scraping library used to extract data from HTML and XML files. It makes it easy to navigate and search through the document tree.

```
pip install beautifulsoup4
```

- **Pyshorteners:**

Pyshorteners is a simple library for generating shortened URLs using popular URL-shortening services like Bit.ly and TinyURL.

```
pip install pyshorteners
```

- **NetworkX:**

NetworkX is a library for creating, analyzing, and visualizing complex networks (graphs). It is often used for graph-based algorithms and network analysis.

```
pip install networkx
```

- **Tkinter:**

Tkinter is a standard GUI library in Python. It provides tools for building desktop applications with a graphical user interface, including buttons, menus, and forms. It is built into Python so no need for installation.

- **Smtplib:**

Smtplib is a built-in Python library used to send emails using the Simple Mail Transfer Protocol (SMTP). It allows you to construct and send email messages programmatically.

- **Email.mime:**

Email.mime is a built-in module used to create MIME-based email messages. It enables the inclusion of attachments, text, and other components in an email.

- **Time:**

Time is a built-in Python module for manipulating and tracking time-related operations like measuring execution time and implementing delays.

Deliverables

Each group shall submit the project accompanied by a report in soft form as a PDF. Pack everything into a single folder (i.e. your report and code file(s)) and name it as your session and roll number, for example, **"(2023)01_02_03_04"** or **"(2021)01_(2022)02_(2023)03_04"**.

Grading Scheme

This project comprises 30% of this subject. The grading scheme is summarized here:

Rubric	**[CLO-5]**		**Use of OOP [CLO-2]**	**Knowledge of Advanced OOP techniques [CLO-4]**	**Report [CLO-6]**	**Presentation [CLO-7]**
	Input Dataset Collection Collect the data that have to be passed on to the code.	**Customization and Adaptation to the Project** Design the framework by importing the required libraries, and importing the input in the code				

		for the project requirements.				
Marks	2	8	5	3	5	7

Rubric Details:

Dataset Collection	Source of Input dataset (2)		
Customization and Adaptation to the Project	Import necessary libraries. (3)	Load the data, ensuring it's in a suitable format for analysis. (5)	
Use of OOP	Just the primitive use. (1)	Apply advanced principles like inheritance, polymorphism, and abstraction to enhance code reusability and scalability. (2)	Good use of multiple classes and their relations. (2)
Knowledge of Advanced OOP techniques	Clear understanding of how the chosen library for a specific project, including its underlying algorithm, assumptions, strengths, and limitations (1)	Understand the concepts of GUI (2)	
Report	No or Incomplete Report Submitted (0)	A good Report but the results could be presented in a better way like in the form of a GIF. (3)	A good quality report with proper representation of the results. (5)
Presentation	No or Incomplete Presentation (0)	Adequate Presentation but Needs Improvement. (4)	Excellent Presentation with Visual Appeal (7)

Scholastic Ethics

It is *emphasized* that each group should submit their work.
Plagiarism will **not** be tolerated.
Remember copy-paste is the worst form of plagiarism and would be treated with the worst form of punishment.

Important dates

Project submission (Section-A): **December-26-2024**

Project submission (Section-B): **December-27-2024**

The project viva date will be conducted on the project submission. At the time of viva, you must have the report in hard form, the folder in soft form, and the running code.

Bibliography

[1] "OpenCV," [Online]. Available: https://docs.opencv.org/4.x/index.html.

[2] "scikit-learn 1.4.1.post1," 16 February 2024. [Online]. Available: https://pypi.org/project/scikit-learn/.

[3] [Online]. Available: https://pypi.org/project/scikit-learn/.