PROGRAM BOOK FOR

SEMESTER INTERNSHIP

Name of the Student - SHAIK ARSHAD WASIB

Name of the College - KAKARAPARTI BHAVANARAYANA COLLEGE

Registration Number - K2001041

Period of Internship - From: - 09/Jan/2023 To: - 10/May/2023

Name & Address of Internship - VXL IT SOLUTIONS, Vijayawada Rd, Beside Big-C, Sriram nagar, Benz circle, Vijayawada, Andhra Pradesh 520010.

KAKARAPARTI BHAVANARAYANA COLLEGE

(AUTONOMOUS)

Affiliated to Krishna University

YEAR: 2020 - 2023

An Internship Report on

"A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN DEGREE PROGRAMS"

Submitted in accordance with the requirement for the degree of

BACHELOR OF COMPUTER APPLICATIONS

Under the Faculty Guideship of

Mr. M. SANTHI BABU

Department of **Computer Science & Applications,**

KBN College (Autonomous)

Submitted by:

SHAIK ARSHAD WASIB

Reg. No: **K2001041**

Department of Computer Science & Applications

KAKARAPARTI BHAVANARAYANA COLLEGE

(AUTONOMOUS)

Instructions to Students

Please read the detailed Guidelines on Internship hosted on the website of AP State Council of Higher Education **https://apsche.ap.gov.in**

1. It is mandatory for all the students to complete Semester internship either in V Semester or in VI Semester.
2. Every student should identify the organization for internship in consultation with the College Principal/the authorized person nominated by the principal.
3. Report to the intern organization as per the schedule given by the College. You must make your own arrangements for transportation to reach the organization.
4. You should maintain punctuality in attending the internship. Daily attendance is compulsory.
5. You are expected to learn about the organization, policies, procedures, and processes by interacting with the people working in the organization and by consulting the supervisor attached to the interns.
6. While you are attending the internship, follow the rules and regulations of the intern organization.
7. While in the intern organization, always wear your College Identity Card.
8. If your college has a prescribed dress as uniform, wear the uniform daily, as you attend to your assigned duties.
9. You will be assigned a Faculty Guide from your College. He/She will be creating a WhatsApp group with your fellow interns. Post your daily activity done and/or any difficulty you encounter during the internship.
10. Identify five or more learning objectives in consultation with your Faculty Guide. These learning objectives can address:
    a. Data and Information you are expected to collect about the organization and/or industry.
    b. Job Skills you are expected to acquire.
    c. Development of professional competencies that lead to future career success.
11. Practice professional communication skills with team members, co-interns, and your supervisor. This includes expressing thoughts and ideas effectively through oral, written, and non-verbal communication, and utilizing listening skills.

12. Be aware of the communication culture in your work environment. Follow up and communicate regularly with your supervisor to provide updates on your progress with work assignments.

    Never be hesitant to ask questions to make sure you fully understand whatyou need to do your work and to contribute to the organization.

13. Be regular in filling up your Program Book. It shall be filled up in your own handwriting. Add additional sheets wherever necessary.
14. At the end of internship, you shall be evaluated by your Supervisor of the intern organization.
15. There shall also be evaluation at the end of the internship by the Faculty Guide and the Principal.
16. Do not meddle with the instruments/equipment you work with.
17. Ensure that you do not cause any disturbance to the regular activities of the intern organization.
18. Be cordial but not too intimate with the employees of the intern organization and your fellow interns.
19. You should understand that during the internship program, you are the ambassador of your college, and your behavior during the internship program is of utmost importance.
20. If you are involved in any discipline related issues, you will be withdrawn from the internship program immediately and disciplinary action shall be initiated.
21. Do not forget to keep up your family pride and prestige of your college.

.....<<@>>.....

Student's Declaration

I'm **Mr. SHAIK ARSHAD WASIB** student of **Bachelor Of Computers Applications** Program, Reg. No. **K2001041** of the Department of **Computer Science & Applications**, Kakaraparthi Bhavanarayana College (Autonomous) do hereby declare that I have completed the mandatory internship from **09-Jan-2023** to **10-May-2023** in **VXL IT Solutions** in Vijayawada under the Faculty Guideship of Honorable **Mr. M. SANTHI BABU**, Department of **Computers Science & Applications, Kakaraparthi Bhavanarayana College (Autonomous).**

(Signature and Date)

Official Certification

This is to certify that I'm **Mr. SHAIK ARSHAD WASIB Reg. No: K2001041** has completed his/her Internship in **VXL IT Solutions** on '**A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN DEGREE PROGRAMS'** under my supervision as a part of partial fulfillment of the requirement for the Degree of **Bachelor Of Computers Applications** in the Department of **Computer Science & Applications, "Kakaraparthi Bhavanarayana College" (Autonomous).**

This is accepted for evaluation

(Signatory with Date Seal)

Endorsements

Faculty Guide

Head of the Department

Principal

Certificate from Intern Organization

This is to certify that **Mr. SHAIK ARSHAD WASIB** Reg. No **K2001041** of **Kakaraparthi Bhavanarayana College (Autonomous).** Underwent internship in **VXL IT Solutions** from **09-Jan-2023** to **10-May-2023**.

The overall performance of the intern during his/her internship is found to be Satisfactory (Satisfactory/Not Satisfactory).

(Authorized Signatory with Date and Seal)

Acknowledgments

This project work is a golden opportunity for learning and self-development. We consider our self very lucky to have so many people lead us through in completion of this project.

We express our sincere thanks to the Management and our beloved Principal **Dr. V. NARAYANA RAO** for providing such wonderful facilities required encouragement in completion of this entire project work.

We place a Record, Our sincere gratitude to **Sri. P. RAVINDRA**, HOD of Computer Applications for his constant encouragement. He monitored the progress and arranged all facilities to make this project easier. We owe our profound gratitude to Project Guide **Mr. M. SANTHI BABU** who took keen interest on this project work and guided us all along and whose patience we have portably used to the limit. She was always so involved in the entire process, shared her knowledge, and encouraged to think.

We would like to thank all the other faculty members, technical staff and supporting staff who have provided their contribution in the completion of this project work.

Name	Roll. No.	Reg. No.
SHAIK ARSHAD WASIB	203641	K2001041

Contents

S. No	Index	Page No.
1	**Chapter-1 (Executive Summary)**	1 - 3
2	**Chapter-2 (Overview of the organization)**	4 - 8
3	**Chapter -3 (Internship Part)**	9 - 11
4	Activity's Log	12 - 87
5	Outcomes Description	88 - 92
6	Student Self Evaluation of the Short-Term Internship	93
7	Evaluation by the Supervisor of the Intern Organization	94
8	Photos & Video Links	95 - 98
9	Evaluation	99 - 100
10	Marks Statement	101 - 102

CHAPTER 1: EXECUTIVE SUMMARY

1.1 Introduction:

The internship report shall have a brief executive summary. It shall include five or more Learning Objectives and Outcomes achieved, a brief description of the sector of business and **VXL IT Solutions** and summary of all the activities done by the intern during the period.

This executive summary provides an overview of a machine learning approach developed to track and predict student performance in degree programs. The objective is to leverage the power of machine learning algorithms to analyze student data and generate actionable insights that can aid in identifying at-risk students, improving retention rates, and enhancing overall program outcomes.

Methodology:

The machine learning approach combines data from various sources, including student demographic information, academic records, course grades, and engagement metrics. These data points are collected and processed to create a comprehensive dataset for analysis. The methodology involves several keys:

steps:

Data Collection: Gathering relevant data from institutional databases, learning management systems, and other sources to establish a comprehensive dataset.

Data Preprocessing: Cleaning and transforming the data to ensure consistency and compatibility for machine learning algorithms. This includes handling missing values, normalizing data, and encoding categorical variables.

Feature Selection: Identifying the most informative features that have a significant impact on student performance. This step helps reduce dimensionality and improve model accuracy.

1.2 Learning Objectives:

- Understand the fundamentals of machine learning and its applications in the field of education.
- Learn the data collection and preprocessing techniques necessary to create a comprehensive dataset for tracking and predicting student performance.
- Gain knowledge of feature selection methods to identify the most relevant and informative factors affecting student performance.
- Explore various machine learning algorithms, such as decision trees, random forests, or neural networks, and understand their suitability for predicting student performance.

- Understand the importance of early identification of at-risk students and explore how machine learning can aid in timely intervention strategies to support struggling students.
- Gain insights into the development of personalized academic intervention strategies based on the analysis of student data and machine learning predictions.

Description about the organization:

VXL IT Solutions is a leading global technology consulting and software services organization. The company provides a wide range of IT solutions and services to clients across various industries, including finance, healthcare, retail, and manufacturing. VXL IT Solutions has established a strong reputation for delivering high-quality and innovative solutions to meet the evolving needs of its clients.

Organizational Structure:

VXL IT Solutions follows a well-defined organizational structure designed to promote efficiency, collaboration, and effective project management. The structure typically includes the following key

components:

1. **Leadership Team**: The organization is led by a team of experienced executives who provide strategic direction and oversee the overall operations of the company. This team is responsible for setting business goals, driving growth initiatives, and ensuring client satisfaction.
2. **Business Units or Departments:** VXL IT Solutions is divided into various business units or departments based on service offerings, industry focus, or technology specialization. This structure allows for efficient resource allocation, domain expertise, and targeted client engagement. Common departments may include software development, consulting, quality assurance, and support.
3. **Project Teams:** Within each department, project teams are formed to execute specific client engagements or internal initiatives. These teams consist of skilled professionals with diverse expertise, including project managers, software developers, business analysts, and quality assurance engineers. Project teams collaborate closely with clients to understand their requirements, develop solutions, and deliver results within the defined timelines and budgets.
4. **Support Functions:** VXL IT Solutions also maintains support functions that provide essential services to the organization and its projects. These functions may include human resources, finance and accounting, marketing, and IT infrastructure support. They ensure the smooth functioning of the organization and support the day-to-day operations.

5. **Centers of Excellence:** VXL IT Solutions may establish Centers of Excellence (CoEs) to foster innovation and promote expertise in specific technology domains or industry verticals. CoEs bring together subject matter experts who drive research and development initiatives, create best practices, and provide thought leadership within their respective areas of focus.

1.3 Summary of activities done:

- **Data collection:** Gather relevant data about students, including their academic records, demographic information, course enrollment, grades, and any other available data points that may impact student performance.

- **Data preprocessing:** Clean the collected data by removing inconsistencies, handling missing values, and transforming the data into a suitable format for analysis. This step may also involve feature engineering, where new features are created from the existing data to enhance the predictive power of the model.

- **Feature selection:** Identify the most relevant features that have a significant impact on student performance. This can be done through statistical analysis or feature selection techniques such as correlation analysis, information gain, or regularization methods.

- **Model selection:** Choose an appropriate machine learning model for the task of predicting student performance. This could involve various algorithms such as decision trees, random forests, support vector machines (SVM), logistic regression, or neural networks.

- **Model training:** Split the data into training and validation sets. Use the training set to train the chosen machine learning model by feeding it with the input features and corresponding target variable (e.g., final grades). The model learns patterns and relationships within the data during this process.

CHAPTER 2: OVERVIEW OF THE ORGANIZATION

2.1 Introduction:

- VXL IT Solutions is an information technology consulting and services company that provides a range of IT solutions to clients worldwide. Here is an overview of the key aspects of the organization:
- **Services:** VXL IT Solutions offers a range of services that include IT consulting, software development, cloud services, big data analytics, digital transformation, and cybersecurity. They also provide customized solutions to meet specific client needs.
- **Clients:** VXL IT Solutions serves clients across various industries, including healthcare, finance, education, retail, and manufacturing. They work with both small and large organizations, including Fortune 500 companies.
- **Team:** VXL IT Solutions has a team of experienced professionals with expertise in various areas of IT, including software development, data analytics, cybersecurity, and cloud services. The team comprises developers, analysts, project managers, and consultants who work together to deliver high-quality solutions.
- **Technology Partners:** VXL IT Solutions partners with leading technology providers such as Microsoft, AWS, and Google to provide cutting-edge solutions to clients. They leverage the latest technologies and tools to deliver innovative solutions that meet client needs.
- **Quality Standards:** VXL IT Solutions follows rigorous quality standards and processes to ensure the delivery of high-quality solutions. They are ISO 9001:2015 certified and adhere to industry standards and best practices.

2.2 Here are the vision and mission statements of VXL IT Solutions

Vision and Mission Statements of VXL IT Solutions

Vision statement:

"To be a trusted global partner, driving digital transformation and empowering businesses through innovative technology solutions."

Mission Statement:

"We are dedicated to delivering exceptional IT consulting and services, leveraging our expertise, industry knowledge, and customer-centric approach to help organizations achieve their strategic goals. Through collaboration, innovation, and a commitment to excellence, we aim to provide transformative technology solutions that drive growth and deliver tangible business value."

To achieve the mission of VXL IT Solutions, the company focuses on the following key objectives:

- **Client Satisfaction:** Ensuring client satisfaction is a top priority for VXL IT Solutions. The company strives to understand and exceed client expectations by delivering high-quality solutions, providing excellent customer service, and fostering long-term partnerships.
- **Innovation and Technological Excellence:** VXL IT Solutions aims to stay at the forefront of technological advancements and industry trends. By fostering a culture of innovation, the company encourages its employees to explore new technologies, develop expertise in emerging areas, and deliver cutting-edge solutions to clients.
- **Value Creation:** VXL IT Solutions is committed to creating value for its clients by delivering solutions that drive tangible business outcomes. The company focuses on understanding client needs, identifying areas for improvement, and providing tailored solutions that optimize processes, enhance efficiency, and generate measurable results.
- **Employee Development and Engagement:** The company recognizes that its employees are its greatest asset. VXL IT Solutions is dedicated to nurturing a supportive and collaborative work environment, promoting employee growth and development, and fostering a strong sense of engagement and ownership among its workforce.
- **Ethical and Responsible Practices:** VXL IT Solutions upholds the highest ethical standards and follows responsible business practices. The company is committed to maintaining data privacy and security, ensuring compliance with regulatory requirements, and conducting business in an ethical and transparent manner.

2.3 Value

Our heritage as a world leader in thin-client, cloud and virtualised computing gives us a unique insight into the needs of software users – and has underpinned the creation of our outstanding range of software. This includes our latest Windows and Linux hybrid working solutions, **Fusion SecureDesktop** and **CloudDesktop On the Go**, our **Fusion UEM**, a unified endpoint management software, which makes mobile and desktop device management easier and more affordable; **CloudDesktop**, which quickly and easily repurposes PCs as thin clients – and **Illumineye DS** (Digital Signage), a powerful and easy-to-use digital signage software and media player solution.

2.4 Policy of the Organization, in relation to the intern role:

1. **Code of Conduct:** Organizations usually have a code of conduct that outlines the behavior and ethical standards expected of their developers. This may include guidelines on professionalism, collaboration, communication, and respect for others.
2. **Intellectual Property:** Organizations typically have policies around intellectual property that outline ownership and rights related to code and other software assets created by developers. This may include guidelines on the use of open-source software, licensing agreements, and copyright laws.
3. **Security:** Organizations typically have policies around security that outline best practices for protecting software and data. This may include guidelines on data privacy, encryption, authentication, and access control.
4. **Quality Assurance:** Organizations typically have policies around quality assurance that outline standards and procedures for testing, bug tracking, and software deployment. This may include guidelines on code review, automated testing, and continuous integration and deployment.
5. **Continuous Learning:** Organizations typically encourage their developers to pursue ongoing training and education to stay up-to-date with the latest technologies and trends in their field. This may include opportunities for attending conferences, participating in training programs, and pursuing certifications.

2.5 Organizational Structure:

VXL IT Solutions follows a well-defined organizational structure designed to promote efficiency, collaboration, and effective project management. The structure typically includes the following key

components:

1. **Leadership Team:** The organization is led by a team of experienced executives who provide strategic direction and oversee the overall operations of the company. This team is responsible for setting business goals, driving growth initiatives, and ensuring client satisfaction.
2. **Business Units or Departments:** VXL IT Solutions is divided into various business units or departments based on service offerings, industry focus, or technology specialization. This structure allows for efficient resource allocation, domain expertise, and targeted client engagement. Common departments may include software development, consulting, quality assurance, and support.

3. **Project Teams:** Within each department, project teams are formed to execute specific client engagements or internal initiatives. These teams consist of skilled professionals with diverse expertise, including project managers, software developers, business analysts, and quality assurance engineers. Project teams collaborate closely with clients to understand their requirements, develop solutions, and deliver results within the defined timelines and budgets.

4. **Support Functions:** VXL IT Solutions also maintains support functions that provide essential services to the organization and its projects. These functions may include human resources, finance and accounting, marketing, and IT infrastructure support. They ensure the smooth functioning of the organization and support the day-to-day operations.

5. **Centers of Excellence:** VXL IT Solutions may establish Centers of Excellence (CoEs) to foster innovation and promote expertise in specific technology domains or industry verticals. CoEs bring together subject matter experts who drive research and development initiatives, create best practices, and provide thought leadership within their respective areas of focus.

2.6 Roles and responsibilities of the employees in which the intern is placed.

Collaborating with senior developers to develop and implement software solutions.Writing clean and efficient code according to industry standards and best practices.Participating in design and code reviews to ensure the code quality and adherence to specifications.

Assisting in troubleshooting and debugging issues in software applications.

Learning new programming languages, tools, and technologies to stay updated with the latest trends.

Creating and maintaining technical documentation for software development projects.

Communicating effectively with other team members and stakeholders to gather requirements and provide updates on the project's progress.

Following the organization's security policies and standards to ensure the safety of data and systems.

Maintaining a positive attitude and demonstrating a willingness to learn and grow as a developer.

Maintaining professionalism and adhering to the company's code of conduct and ethical standards.

2.7. Future Plans of the Organization.

As per the information available, VXL IT Solutions is a growing organization that aims to continue expanding its reach and impact in the education sector. Some of the future plans of the organization include:

1. **Expansion into new geographic regions:** The organization aims to expand its software solutions to more schools and educational institutions in different regions across the world.
2. **New product development:** The organization plans to develop and launch new software solutions and features that cater to the changing needs of the education sector.
3. **Strategic partnerships:** The organization is keen on partnering with other companies and organizations in the education sector to leverage their strengths and create new opportunities.
4. **Research and development:** The organization plans to invest more in research and development to stay at the forefront of innovation in the education technology space.
5. **Customer support and service:** The organization recognizes the importance of providing excellent customer support and service to its clients, and plans to continue improving in this area.

CHAPTER 3: INTERNSHIP PART

3.1 Activities and Responsibilities

Description of the Activities/Responsibilities in the Intern Organization during Internship, which shall include - details of working conditions, weekly work schedule, equipment used, and tasks performed. This part could end by reflecting on what kind of skills the intern acquired.

Creating the **'A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN DEGREE PROGRAMS'** by VXL IT Solutions involved various activities and responsibilities. The software developers and designers had to work together to ensure the successful completion of the project. This section will discuss in detail the working conditions, weekly work schedule, equipment used, and tasks performed during the development of the **A machine learning approach for tracking and predicting student performance in degree programs.**

Working Conditions:

The development of the **A machine learning approach for tracking and predicting student performance in degree programs** required the use of a computer and other software development tools. The developers had to work in a comfortable and conducive environment that promoted creativity and productivity. The work environment was designed to reduce distractions and ensure that the developers had minimal interruptions.

Weekly Work Schedule:

The developers and designers had to follow a structured weekly work schedule that allowed for effective time management and timely completion of the project. The weekly work schedule included planning and review meetings, individual tasks, and team-based activities. The developers had to adhere to the schedule to ensure that the project was completed on time.

Equipment Used:

To develop a machine learning approach for tracking and predicting student performance in degree programs, the following equipment used:

1. **Computer hardware:** A powerful computer with high-performance processors and graphics card may be required to handle large datasets and complex machine learning algorithms.
2. **Programming languages and software:** The machine learning approach require programming languages such as Python, Machine Learning, Deep Learning along with software libraries such as Scikit-learn, Tensorflow, or Keras, tkinter(tk), numpy, pandas, matplotlib for data analysis, modeling, and visualization.

3. **Data collection tools:** Tools such as Learning Management Systems (LMS), student information systems, and educational data mining software may be used to collect and store data on student performance, behavior, and other relevant metrics.
4. **Datasets:** Student Datasets are required for training and testing the datasets for to prepare the model for our project using python.
5. **Hardware Requirements:** Processor-Intel Core i3/i5/i7/i9, Ram-4GB/8GB/16GB, Hard Disk-500 GB.
6. **Software Requirements:** Operating System-Windows 8, Windows 10, Windows 11, Coding Language-Python, Documentation-MS Office.

Tasks Performed:

A machine learning approach for tracking and predicting student performance in degree programs involves various tasks, including:

1. **Data collection:** Collecting data on student performance, behavior, and other relevant metrics from multiple sources such as Learning Management Systems (LMS), student information systems, and educational data mining software.

2. **Data preprocessing:** Cleaning, transforming, and preparing data for analysis and modeling, which involves tasks such as data cleaning, feature extraction, normalization, and data augmentation.

3. **Data analysis and modeling:** Analyzing the prepared data to identify patterns, trends, and correlations using machine learning algorithms such as classification, regression, and clustering. This task also involves model selection, hyperparameter tuning, and training the models on the data.

4. **Model evaluation and validation:** Evaluating the performance of the machine learning models using metrics such as accuracy, precision, recall, and F1-score. This task involves using techniques such as cross-validation, bootstrapping, and A/B testing to ensure the models are accurate and reliable.

5. **Prediction and recommendation:** Using the trained models to predict student performance and behavior, provide personalized recommendations for study plans, courses, and support services based on individual needs and preferences.

3.2 Skills Acquired:

A machine learning approach for tracking and predicting student performance in degree programs requires a set of skills and expertise. Some of the skills that can be acquired by working on such a project are:

1. **Data analysis and preprocessing:** The ability to clean, transform, and prepare data for analysis is a critical skill in machine learning. Working on this project can help acquire skills in data analysis, feature selection, and normalization.

2. **Machine learning algorithms:** The project involves working with various machine learning algorithms, including **regression**, **classification**, and **clustering**, '**Random Forest Algorithm'**, '**SVM Algorithm'**, '**Logistic Regression Algorithm'**, '**Propose Ensemble-based Progressive Prediction (EPP) Algorithms'**. '**Predict Performance'** to upload student on going test marks and to **predict GPA for future course**. Acquiring skills in these algorithms can help in understanding the strengths and limitations of each algorithm and selecting the appropriate algorithm for the task at hand.

3. **Programming skills:** Programming languages such as Python are commonly used in machine learning projects. Acquiring programming skills in these Language can help in building and deploying machine learning models.

4. **Statistics and mathematics:** Machine learning involves working with statistical models and mathematical concepts such as linear algebra, probability, and calculus. Acquiring these skills can help in understanding the underlying concepts of machine learning algorithms and their applications.

5. **Critical thinking and problem-solving:** Developing a machine learning approach for tracking and predicting student performance in degree programs involves critical thinking and problem-solving skills. It requires the ability to analyze and interpret data, identify patterns, and develop strategies for improving student performance.

ACTIVITY LOG FOR THE FIRST WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 09-Jan-2023	**Learned the introduction of 'Python Full Stack Development'**	Learned the introduction of Front-end programming Languages such as HTML,CSS,JavaScript, Bootstrap, AngularJS 8, TypeScript.	
Day – 2 10-Jan-2023	**Learned the introduction of Back-end programming Languages**	Introduction of Python, Django, Flask.	
Day – 3 11-Jan-2023	**Learned the introduction of HTML**	Hyper Text Markup Language, Types of Webpages, Types of Markup Languages, Types of tags.	
Day – 4 12-Jan-2023	**HTML Editors**	Visual Studio Code,Sublime,EditPlus, Notepad++,Brackets, Komodo Edit, CoffeeCup,Adobe Dreamweaver	
Day – 5 13-Jan-2023	**Structure of HTML Program**	Developing of HTML Program	
Day – 6 14-Jan-2023	**HOLIDAY**	An Account of **"BHOGI"** Festival	

WEEKLY REPORT

WEEK – 1 From Date (09-01-2023) to Date (14-01-2023)

During the first week of learning introduction of 'Python Full Stack Development', I started by getting introduction of front-end programming languages such as **HTML, CSS, JavaScript, Bootstrap, AngularJS 8, TypeScript.**

HTML:

HTML program is a collection of markup code written in HTML (**Hypertext Markup Language**). HTML is the standard language used to create web pages on the internet. The program consists of a series of HTML tags, which are used to define the structure and content of the web page.

Overall, an HTML program serves as a blueprint for a web page, defining its structure and content. When opened in a web browser, the browser interprets the HTML code and renders the page accordingly, displaying the text, images, and interactive elements as intended by the developer."

CSS:

"A CSS program is a collection of rules written in CSS (Cascading Style Sheets), which is a style sheet language used to describe the presentation and visual style of an HTML document. CSS is essential for controlling the layout, colors, fonts, and other visual aspects of web pages.

```
p {
  font-family: Arial, sans-serif;
  color: #333333;
}
```

CSS programs can also include media queries, which allow you to apply different styles based on the device or screen size. This enables responsive web design, where the layout and appearance of a web page can adapt to different screen sizes, such as desktops, tablets, and mobile devices.

When a web browser renders an HTML document, it interprets the CSS rules and applies the specified styles to the corresponding HTML elements, resulting in the desired visual presentation of the web page."

JAVASCRIPT:

A JavaScript program is a collection of instructions written in the JavaScript programming language. JavaScript is a versatile and widely used scripting language that allows developers to create interactive and dynamic elements on web pages.

In a JavaScript program, you can define variables, functions, loops, conditional statements, and other constructs to manipulate and control the behavior of web pages. JavaScript can be embedded within HTML documents using the `<script>` tag or stored in separate JavaScript files and linked to HTML documents.

javascript

```
let name = prompt("What is your name?");
let greeting = "Hello, " + name + "!";
alert(greeting);
```

JavaScript is a powerful programming language that enables developers to create interactive features, validate forms, perform calculations, manipulate data, and much more. It plays a crucial role in enhancing the functionality and interactivity of web pages, making them more engaging and responsive to user interactions.

BOOTSRAP:

A Bootstrap program refers to a web development framework called Bootstrap, which is a popular front-end framework used to create responsive and visually appealing websites and web applications. Bootstrap provides a set of pre-built HTML, CSS, and JavaScript components and styles that can be easily customized and integrated into web projects.

ANGULAR JS:

An AngularJS 8 program refers to a web application developed using the AngularJS framework version 8. AngularJS is a JavaScript-based framework maintained by Google that enables the creation of dynamic and interactive web applications.

Two-Way Data Binding: AngularJS offers powerful two-way data binding, which means any changes made to the data in the Model or View are automatically synchronized, keeping them in sync. This simplifies data manipulation and reduces the need for manual updates.

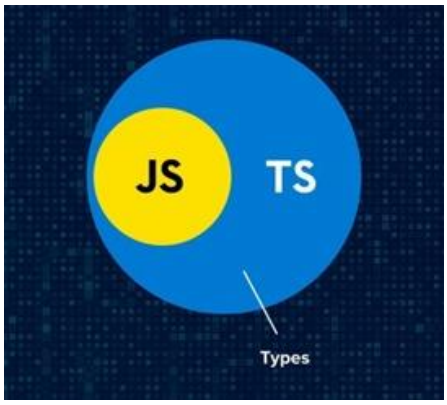
Directives: AngularJS introduces custom HTML attributes called directives. Directives enable the creation of reusable components and allow developers to extend HTML with new behaviors and functionality. AngularJS includes built-in directives like `ng-repeat`, `ng-model`, `ng-show`, etc.

Dependency Injection: AngularJS employs dependency injection, which helps manage and organize dependencies between different components of the application. This allows for better modularity, testability, and maintainability of the codebase.

Templating: AngularJS provides a powerful templating engine that allows developers to create dynamic HTML templates using AngularJS expressions, filters, and directives. These templates are rendered and updated automatically as the underlying data changes.

Routing: AngularJS offers a built-in routing mechanism that enables the creation of single-page applications (SPAs) with multiple views. Developers can define routes and associate them with specific controllers and templates to create a seamless user experience.

TYPESCRIPT:



A TypeScript program is a collection of code written in the TypeScript programming language. TypeScript is a superset of JavaScript that introduces static typing and additional features to enhance JavaScript development.

Static Typing: TypeScript introduces static typing, allowing developers to specify types for variables, function parameters, and return values. This helps catch potential type-related errors during the development phase and improves code quality.

Object-Oriented Features: TypeScript supports object-oriented programming (OOP) concepts such as classes, interfaces, inheritance, and modules. These features enable developers to write more structured and modular code, making it easier to manage complex applications.

ECMA Script Compatibility: TypeScript is a superset of JavaScript, which means that any valid JavaScript code is also valid TypeScript code. TypeScript follows the ECMAScript standard closely and provides support for modern JavaScript features, even before they are widely adopted.

Enhanced Tooling: TypeScript comes with a rich set of tools and features that aid in the development process. It includes a compiler that transpiles TypeScript code to plain JavaScript, providing browser compatibility. Integrated Development Environments (IDEs) and code editors offer enhanced code navigation, autocompletion, and error checking for TypeScript projects.

Type Inference: TypeScript has a powerful type inference system that can automatically infer the types of variables based on their initial values. This reduces the need for explicit type annotations and makes the code more concise and readable.

Introduction of Python

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- Web development (server-side),
- Software development,
- Mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

• Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

• Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know:

• The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

Example Program:

```
print("HelloWorld!")
```

Introduction of Django

What is Django?

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that you can concentrate on building your web applications.

Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

Django is especially helpful for database driven websites.

How does Django Work?

DJANGO FRAMEWORK:

Django follows the MVT design pattern (Model View Template).

• Model - The data you want to present, usually data from a database.

• View - A request handler that returns the relevant template and content - based on the request from the user.

• Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

Model:

The model provides data from the database.

In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.

Template:

A template is a file where you describe how the result should be represented.

Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on .html files.

Django uses standard HTML to describe the layout, but uses Django tags to add logic:

<h1>My Homepage</h1>

My name is {{ firstname }}.

The templates of an application is located in a folder named templates.

URLs:

Django also provides a way to navigate around the different pages in a website.

When a user requests a URL, Django decides which view it will send it to.

This is done in a file called urls.py.

So, what is Going on?:

When you have installed Django and created your first Django web application, and the browser requests the URL, this is basically what happens:

1. Django receives the URL, checks the urls.py file, and calls the view that matches the URL.
2. The view, located in views.py, checks for relevant models.
3. The models are imported from the models.py file.
4. The view then sends the data to a specified template in the template folder.
5. The template contains HTML and Django tags, and with the data it returns finished HTML content back to the browser.

Django History

Django was invented by Lawrence Journal-World in 2003, to meet the short deadlines in the newspaper and at the same time meeting the demands of experienced web developers.

Initial release to the public was in July 2005.

Latest version of Django is 4.0.3 (March 2022).

Introduction of Flask

Python Flask

Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals.

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO).

What is Flask?

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

What is WSGI?

It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.

What is Jinja2?

Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

Flask Environment Setup

To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

Install virtual environment (virtualenv)

Introduction of HTML

HTML program is a collection of markup code written in HTML (Hypertext Markup Language). HTML is the standard language used to create web pages on the internet. The program consists of a series of HTML tags, which are used to define the structure and content of the web page.

HTML Syntax:

1. The HTML document starts with a `<!DOCTYPE html>` declaration that tells the browser what type of document to expect.

2. The HTML document is wrapped in `<html>` tags, and contains two main sections: the `<head>` section and the `<body>` section.

3. The `<head>` section contains information about the document, such as the title and any stylesheets or scripts that are needed. This section is not displayed on the page.

4. The `<body>` section contains the content of the document that is displayed on the page. This section can contain a variety of HTML elements, such as headings, paragraphs, lists, images, links, and more.

Types of Webpages:

There are several types of web pages that can be created using HTML, some of which include:

- Static Web Pages: These are basic web pages that do not change based on user input or actions. They are often used for displaying information that does not need to be updated frequently.
- Dynamic Web Pages: These are web pages that change based on user input or actions. They are often used for web applications that require user interaction. Here is an example of a dynamic web page that displays a random number when a button is clicked:
- Responsive Web Pages: These are web pages that are designed to adapt to different screen sizes, such as those on mobile devices or desktop computers. They are often created using CSS media queries.

Types of Tags:

HTML tags are used to define the structure and content of a web page. There are several types of tags in HTML, including:

1. **Structural Tags:** These tags are used to define the basic structure of a web page, such as headings, paragraphs, and lists. Examples include `<h1>`, `<p>`, `<ul>`, and `<ol>`.
2. **Formatting Tags:** These tags are used to define the appearance of text and other elements on a web page, such as bold or italicized text. Examples include `<b>`, `<i>`, and `<u>`.
3. **Link Tags:** These tags are used to create hyperlinks between web pages or to other resources, such as images or files. Examples include `<a>` and `<img>`.
4. **Input Tags:** These tags are used to create interactive elements on a web page, such as forms and buttons. Examples include `<input>`, `<button>`, and `<select>`.
5. **Semantic Tags:** These tags are used to provide additional meaning and context to the content on a web page, which can help with search engine optimization and accessibility. Examples include `<header>`, `<nav>`, and `<footer>`.

HTML EDITORS:

Adobe Dreamweaver: Dreamweaver is a professional-grade HTML editor that is widely used for web development. It offers a range of features, such as a visual editor, code highlighting, and integrated FTP/SFTP support.

Visual Studio Code: Visual Studio Code is a free and open-source code editor that supports HTML, as well as other programming languages. It offers features such as code highlighting, code completion, and integrated debugging.

Structure of HTML Program:

The Document Type Declaration (DTD): This is the first line of an HTML document and tells the web browser which version of HTML is being used.

Example: `<!DOCTYPE html>`

The `<html>` Element: This element contains all the other elements on the page and is the root element of the HTML document.

Example: `<html>...</html>`

The `<head>` Element: This element contains metadata about the HTML document, such as the page title, links to stylesheets and scripts, and other information that is not displayed on the web page.

Example: `<head>...</head>`

The `<title>` Element: This element specifies the title of the web page, which is displayed in the browser's title bar.

Example: `<title>My Web Page</title>`

The `<body>` Element: This element contains the visible content of the web page, such as text, images, and other elements.

Example: `<body>...</body>`

Other HTML Elements: These elements make up the visible content of the web page, such as headings, paragraphs, lists, images, and links.

Example: `<h1>Heading 1</h1>`, `<p>Paragraph text</p>`, `<ul><li>List item 1</li><li>List item 2</li></ul>`

ACTIVITY LOG FOR THE SECOND WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 16-Jan-2023	**HOLIDAY**	An Account of **"KANUMA"** Festival	
Day - 2 17-Jan-2023	**HTML Tags**	Heading level tags, Paragraph tag, Line breaks, Elements	
Day – 3 18-Jan-2023	**Text Formatting tags**	Tag Description	
Day – 4 19-Jan-2023	**Learned the introduction of CSS**	What is CSS?, Why use CSS?, CSS syntax	
Day – 5 20-Jan-2023	**CSS Selectors**	CSS selectors, CSS colors	
Day – 6 21-Jan-2023	**CSS Backgrounds**	CSS backgrounds-color & CSS background Image	

WEEKLY REPORT

WEEK – 2 From Date (16-01-2023) to Date (21-01-2023)

HTML Tags:

HTML provides six levels of headings, from `<h1>` to `<h6>`. The headings are used to structure the content of a web page and provide an outline of the information presented on the page.

Here are the different levels of heading tags in HTML:

1. `<h1>`: This tag is used for the most important heading on the page, usually the title of the page or the main heading of a section.

Example: `<h1>Welcome to My Website</h1>`

2. `<h2>`: This tag is used for the subheadings that follow the main heading and represent major sections of the page.

Example: `<h2>About Us</h2>`

3. `<h3>`: This tag is used for headings that are subordinate to the `<h2>` headings and represent sub-sections of the page.

Example: `<h3>Our Team</h3>`

4. `<h4>`: This tag is used for headings that are subordinate to `<h3>` headings.

Example: `<h4>John Smith</h4>`

5. `<h5>`: This tag is used for headings that are subordinate to `<h4>` headings.

Example: `<h5>Education</h5>`

6. `<h6>`: This tag is used for headings that are subordinate to `<h5>` headings.

Example: `<h6>Bachelor's Degree in Computer Science</h6>`

Paragraph tag:

In HTML, the `<p>` tag is used to define a paragraph of text. It's a block-level element that creates a new line after the paragraph.

Here's an example of how to use the `<p>` tag in HTML:

This is a paragraph of text.

You can use the `<p>` tag to separate blocks of text on your web page and make them easier to read. You can also apply formatting to the text within the `<p>` tag, such as bold or italic text, by using additional HTML tags.

Here's an example of how to use the `<p>` tag with some additional formatting:

This is a paragraph of <u>formatted</u> text.

Line Breaks:

In HTML, you can use the `<br>` tag to insert a line break, which creates a new line without starting a new paragraph. The `<br>` tag is a self-closing tag, which means that it doesn't require a closing tag.

Here's an example of how to use the `<br>` tag in HTML:

```
<p>This is the first line.<br>
This is the second line.</p>
```

In this example, the `<br>` tag is used to create a line break after the first line of text, so that the second line starts on a new line. Note that the `<br>` tag is inserted within the `<p>` tag, so the two lines of text are still part of the same paragraph.

You can also use the `<br>` tag to insert line breaks within other HTML elements, such as headings or list items. Here's an example:

```
<h2>Ingredients:</h2>
<ul>
  <li>Flour</li>
  <li>Sugar</li>
  <li>Eggs<br>
     Milk<br>
     Vanilla extract</li>
  <li>Baking powder</li>
</ul>
```

It's important to use the `<br>` tag sparingly, as it can make your HTML code harder to read and maintain. In general, it's better to use block-level elements like paragraphs and headings to separate content on your web page. However, the `<br>` tag can be useful in certain situations where you need to create a line break without starting a new paragraph.

Elements:

In HTML, an element is a piece of content on a web page that is enclosed in tags. An element can consist of text, images, links, or other types of content. HTML elements are used to structure the content on a web page and provide meaning to the text and other content.

Here's an example of an HTML element:

```
<p>This is a paragraph of text.</p>
```

```
<ul>
  <li>Item 1</li>
  <li>Item 2
    <ul>
      <li>Subitem 1</li>
      <li>Subitem 2</li>
    </ul>
  </li>
  <li>Item 3</li>
</ul>
```

Text Formatting tags:

Tag Description:

In HTML, tags are used to define elements on a web page. Tags are enclosed in angle brackets (<>) and can contain attributes, which provide additional information about the element.

```
<img src="image.jpg" alt="A beautiful landscape">
```

Structural tags: These tags are used to define the structure of the web page, such as headings, paragraphs, lists, and tables. Examples include `<h1>` to `<h6>` for headings, `<p>` for paragraphs, `<ul>` and `<ol>` for unordered and ordered lists, and `<table>` for tables.

Formatting tags: These tags are used to apply formatting to the text and other content on the web page, such as bold or italic text, underlined text, and color. Examples include `<b>` and `<strong>` for bold text, `<i>` and `<em>` for italic text, `<u>` for underlined text, and `<font>` for color.

Link tags: These tags are used to create links to other web pages or resources. Examples include `<a>` for hyperlinks, `<link>` for linking to external stylesheets, and `<img>` for linking to images.

Meta tags: These tags are used to provide additional information about the web page, such as the author, keywords, and description. Examples include `<title>` for the title of the web page, `<meta name="author" content="John Doe">` for the author's name, and `<meta name="keywords" content="HTML, CSS, JavaScript">` for the page's keywords.

Introduction of CSS

What Is CSS?

"A CSS program is a collection of rules written in CSS (Cascading Style Sheets), which is a style sheet language used to describe the presentation and visual style of an HTML document. CSS is essential for controlling the layout, colors, fonts, and other visual aspects of web pages.

Why Use CSS?

CSS is used in web development to separate the content and presentation of a web page, improve consistency and efficiency across multiple pages, provide flexibility in design and layout, and enhance accessibility for users with disabilities.

CSS Syntax:

CSS, or Cascading Style Sheets, uses a simple syntax to define the style and layout of HTML elements on a web page. Here's a brief overview of the CSS syntax:

CSS syntax:h1 {

color: blue;

font-size: 24px;

text-align: center;}

CSS Selectors:

CSS selectors are used to target specific HTML elements on a web page and apply styles to them. Here are some common types of CSS selectors:

Element selectors: Selects elements based on their tag name. For example, `p` selects all `<p>` elements on the page.

Class selectors: Selects elements based on their class attribute. For example, `.intro` selects all elements with a class of "intro".

ID selectors: Selects elements based on their ID attribute. For example, `#header` selects the element with an ID of "header".

Attribute selectors: Selects elements based on the value of their attribute. For example, `[type="text"]` selects all elements with a type attribute of "text".

CSS Colors:

CSS allows web developers to specify a wide range of colors for use in the styling and layout of HTML elements on a web page. Here are some ways to define colors in CSS:

Named colors: CSS includes a set of 147 named colors, such as "red", "green", "blue", and "yellow".

Hexadecimal colors: Colors can be specified using a six-digit hexadecimal code, which represents the amount of red, green, and blue in the color. For example, `color: #FF0000;` sets the text color to red.

RGB colors: Colors can be specified using an RGB (red, green, blue) value, which represents the amount of each color component on a scale from 0 to 255. For example, `color: rgb(255, 0, 0);` sets the text color to red.

HSL colors: Colors can be specified using an HSL (hue, saturation, lightness) value, which allows for more precise control over the color.

For example: `color: hsl(0, 100%, 50%);` sets the text color to a pure red.

CSS also includes other color-related properties, such as opacity, which allows for transparency in elements, and background-color, which sets the background color of an element.

CSS Backgrounds:

CSS background-color:

The `background-color` property in CSS is used to set the background color of an HTML element. This property can be used to set the background color for the entire page or for individual elements, such as `<div>`, `<body>`, or `<header>`.

Here is the basic syntax for using the `background-color` property:

```
selector {
  background-color: value;
}
```

In this syntax, `selector` specifies the HTML element(s) to which the `background-color` property will be applied, and `value` specifies the color to be used for the background.

There are several ways to specify the `value` of the `background-color` property:

Named colors: CSS includes a set of 147 named colors, such as "red", "green", "blue", and "yellow". For example, `background-color: red;` sets the background color to red.

Hexadecimal colors: Colors can be specified using a six-digit hexadecimal code, which represents the amount of red, green, and blue in the color. For example, `background-color: #FF0000;` sets the background color to red.

RGB colors: Colors can be specified using an RGB (red, green, blue) value, which represents the amount of each color component on a scale from 0 to 255. For example, `background-color: rgb(255, 0, 0);` sets the background color to red.

HSL colors: Colors can be specified using an HSL (hue, saturation, lightness) value, which allows for more precise control over the color.

CSS Backgrounds Image:

In addition to setting the `background-color` property, you can also set a background image for an HTML element using the `background-image` property in CSS. This allows you to display a graphic or photo as the background for an element, such as a `<div>` or the entire page.

ACTIVITY LOG FOR THE THIRD WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 23-Jan-2023	**CSS Effects**	CSS Text Overflow, CSS Animations	
Day - 2 24-Jan-2023	**Learned the introduction of JavaScript**	JavaScript can change HTML content, JavaScript Functions and Events	
Day – 3 25-Jan-2023	**JavaScript Syntax**	JavaScript RegExp	
Day – 4 26-Jan-2023	**Holiday**	An account of Republic Day	
Day – 5 27-Jan-2023	**JavaScript Objects**	Object Definitions, Properties, Methods	
Day – 6 28-Jan-2023	**JavaScript Functions**	Function Definition, Parameters, Invocation	

WEEKLY REPORT

WEEK – 3 From Date (23-01-2023) to Date (28-01-2023)

CSS Effects:

CSS Text Overflow:

The `text-overflow` property in CSS is used to control what happens when the text content of an HTML element exceeds the available space.

The `text-overflow` property allows you to specify how the text should be handled when it overflows the available space. There are three possible values for the `text-overflow` property:

`clip`: This value indicates that any text that overflows the available space should be clipped, or cut off, so that it does not display beyond the boundaries of the element.

`ellipsis`: This value indicates that any text that overflows the available space should be replaced with an ellipsis, or a series of three dots (`...`), to indicate that the text has been truncated.

`fade`: This value indicates that any text that overflows the available space should be gradually faded out as it reaches the edge of the element. This can create a visual effect that makes it clear that the content continues beyond the visible area.

Here is an example of how to use the `text-overflow` property to add ellipsis to an HTML element:

```
.selector {
  white-space: nowrap; /* Prevents the text from wrapping */
  overflow: hidden; /* Hides any content that overflows the element */
  text-overflow: ellipsis; /* Replaces the overflowed text with ellipsis */
}
```

CSS Animations:

CSS animations allow you to add movement and visual interest to your web pages. With CSS animations, you can animate any CSS property, such as `color`, `background-color`, `font-size`, `transform`, and many others.

```
@keyframes example-animation {
  from { color: black; }
  to { color: red; }
}
```

Functions: JavaScript functions are used to encapsulate and reuse blocks of code. Functions can take input parameters and return output values, and are declared using the `function` keyword.

Objects: JavaScript objects are used to represent complex data structures and provide a way to organize related data and functionality. Objects can contain properties (which store data values) and methods (which define actions that can be performed on the object).

Conditionals: JavaScript conditionals are used to control the flow of execution based on a condition. Conditionals include the `if`, `else`, and `switch` statements.

Loops: JavaScript loops are used to execute a block of code multiple times. Loops include the `for`, `while`, and `do-while` statements.

Here is an example of some JavaScript code that demonstrates these syntax elements:

```
// Declare a variable
let message = "Hello, world!";
// Define a function
function greet(name) {
  console.log("Hello, " + name + "!");
}
```

JavaScript RegExp:

Regular expressions, or regex, in JavaScript are a powerful tool for pattern matching and text manipulation. Regex allows you to search and match patterns within strings, replace parts of strings, and extract specific information from a string.

Character classes: Character classes allow you to match any character within a specified set. For example, `[abc]` matches any of the characters "a", "b", or "c".

Quantifiers: Quantifiers allow you to specify how many times a character or group of characters should be matched. For example, `a+` matches one or more occurrences of the letter "a".

Anchors: Anchors allow you to match patterns at the beginning or end of a string. For example, `^` matches the beginning of a string, and `$` matches the end of a string.

Grouping: Grouping allows you to group multiple characters or patterns together and apply quantifiers to the entire group. For example, `(ab)+` matches one or more occurrences of the string "ab".

Alternation: Alternation allows you to match one of several patterns. For example, `a|b` matches either the letter "a" or the letter "b".

Here is an example of using regex in JavaScript to match a pattern within a string:

```
const pattern = /hello \w+/;
const str = "Hello world!";
if (pattern.test(str)) {
  console.log("Pattern matched!");
} else {
  console.log("Pattern not matched.");
}
```

JavaScript Objects:

Object Definition:

In JavaScript, an object is a collection of related data and functionality, represented as a set of key-value pairs. Objects can be created using either object literal notation or constructor notation.

Here is an example of defining an object using object literal notation:

```
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 30,
  getFullName: function() {
    return this.firstName + " " + this.lastName;
  }
};
function Person(firstName, lastName, age) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.age = age;
  this.getFullName = function() {
    return this.firstName + " " + this.lastName;
  }
}
const person = new Person("John", "Doe", 30);
```

For example:

```
console.log(person.firstName); // Output: "John"
console.log(person["lastName"]); // Output: "Doe"
console.log(person.getFullName()); // Output: "John Doe"
```

JavaScript Methods and Properties:

JavaScript is a powerful programming language that comes with a lot of built-in methods and properties.

`typeof`: This operator is used to determine the data type of a value or expression. It returns a string that represents the data type, such as `"string"`, `"number"`, `"boolean"`, `"function"`, or `"object"`.

`parseInt` and `parseFloat`: These methods are used to convert a string to an integer or a float, respectively.

`toString`: This method is used to convert a value to a string.

JavaScript Functions:

Function Definition:

In JavaScript, a function is a block of code that performs a specific task. Functions are used to break down large, complex problems into smaller, more manageable ones. They can take input in the form of parameters and return output in the form of a return value.Here is an example of defining a function in JavaScript:

```
function square(x) {
  return x * x;
}
const result = square(5);
console.log(result); // Output: 25
```

Functions can also be defined using function expressions or arrow functions. Here are some examples:

```
const add = function(x, y) {
  return x + y;
}
const multiply = (x, y) => {
  return x * y;}
```

Parameters:

In JavaScript, function parameters are used to pass values to a function when it is called. They are defined inside the function's parentheses and separated by commas.

For example, the following function takes two parameters `x` and `y`:

```
function add(x, y) {
  return x + y;
}
```

For example:

```
const result = add(5, 7);
console.log(result); // Output: 12
```

For example:

```
function greet(name = 'World') {
  console.log(`Hello, ${name}!`);
}greet(); // Output: Hello, World!
greet('Alice'); // Output: Hello, Alice!
```

Invocation: In JavaScript, function invocation refers to the act of calling or executing a function. There are several ways to invoke a function in JavaScript:

1. Function declaration:

```
function add(x, y) {
  return x + y;
}
add(2, 3); // Output: 5
```

2. Function expression:

```
const add = function(x, y) {
  return x + y;
};
add(2, 3); // Output: 5
```

3. Arrow function:

```
const add = (x, y) => {
  return x + y;
};
add(2, 3); // Output: 5
```

ACTIVITY LOG FOR THE FOURTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 30-Jan-2023	**Learned the introduction of Bootstrap**	What is Bootstrap?, Downloading Bootstrap 5	
Day - 2 31-Jan-2023	**Create your first Webpage with Bootstrap5**	Add the HTML 5 doctype, Bootstrap 5 is mobile-first	
Day – 3 01-Feb-2023	**Bootstrap Containers**	Fixed container, Fluid container, Responsive container	
Day – 4 02-Feb-2023	**Bootstrap Colors, Alerts, Buttons**	Text colors & Background colors, Alerts, Button styles	
Day – 5 03-Feb-2023	**Bootstrap Paginations, Navbar, Carousel**	Types of paginations, Types of navbars, Types of carousel	
Day – 6 04-Feb-2023	**Learned the introduction of AngularJs 8**	AngularJs 8 definition, Upgrade angular CLI, Features and Commands.	

WEEKLY REPORT

WEEK – 4 From Date (30-01-2023) to Date (04-02-2023)

Introduction of Bootstrap

What Is Bootstrap?

Bootstrap 5 is a free and open-source CSS framework that provides a collection of pre-built HTML, CSS, and JavaScript components for building responsive and mobile-first web applications. It includes a wide range of features such as layout grids, typography, forms, buttons, navigation, and more, and is designed to simplify front-end web development and speed up the process of creating modern and visually appealing web designs. Bootstrap 5 is the latest version of Bootstrap, released in May 2021, and includes significant updates and improvements over its predecessor, Bootstrap 4.

Bootstrap Syntax:

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Syntax Example</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css">
</head>
<body>
<div class="container mt-4">
<div class="card">
<div class="card-body">
<h5 class="card-title">Card Title</h5>
<p class="card-text">This is a simple card example using Bootstrap.</p>
<a href="#" class="btn btn-primary">Learn More</a>
</div>
</div>
</div>
<!-- Include Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

Downloading Bootstrap 5:

To download Bootstrap 5, you can visit the official Bootstrap website at https://getbootstrap.com/docs/5.0/getting-started/download/

Downloading the compiled and minified CSS, JavaScript, and font files directly from the website. Using a package manager such as npm or yarn to install Bootstrap 5 and its dependencies. Using a CDN to include the Bootstrap 5 files in your project directly from a remote server.

Add the HTML 5 doctype:

To add the HTML5 doctype to a Bootstrap project, you can simply include the following code at the beginning of your HTML file:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>My Bootstrap Project</title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css">
</head>
<body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

Bootstrap Containers:

Fixed Containers:

In Bootstrap 5, fixed containers are a type of container that allow you to center and constrain your content within a fixed-width container. This can be useful when you want to limit the width of your content and prevent it from stretching too wide on larger screens.

`.container-sm` for small screens (less than 576px)

`.container-md` for medium screens (between 576px and 768px)

`.container-lg` for large screens (between 768px and 992px)

`.container-xl` for extra-large screens (between 992px and 1200px)

`.container-xxl` for extra-extra-large screens (above 1200px)

Here's an example of how to create a fixed container that centers and constrains your content to a maximum width of 960 pixels:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bootstrap Fixed Container Example</title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css">
</head>
<body>
<div class="container-lg">
<h1>My Fixed Container</h1>
<p>This content is centered and constrained within a fixed-width container.</p>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

Responsive Containers:

You can use the same responsive grid system as with fixed and fluid containers to create a layout within a responsive container.

Bootstrap Colors:

Text Colors & Background Colors:

In Bootstrap 5, you can easily set the text color and background color of your elements using utility classes.

Text color classes are prefixed with `text-` and can be used on any element to change its color. For example, `text-primary` will set the color of an element to the primary color of your Bootstrap theme. Here's an example:

```
<p class="text-primary">This text is in the primary color</p>
```

Similarly, you can set the background color of an element using the `bg-` prefix. For example, `bg-primary` will set the background color of an element to the primary color of your Bootstrap theme. Here's an example:

```
<div class="bg-primary">
  <p>This div has a primary background color</p>
</div>
```

Additionally, you can combine text color and background color classes to create more complex styles.

For example:

```
<button class="btn btn-success text-white">Click me!</button>
```

Bootstrap Alerts:

Bootstrap alerts are a way to display a short message to the user. They are used to communicate important information or to notify the user of an event. Bootstrap alerts come in different types and can be customized to fit your needs. To use an alert, you can add the `.alert` class to a `<div>` element. The following types of alerts are available in Bootstrap 5:

Primary: `.alert-primary`

Secondary: `.alert-secondary`

Success: `.alert-success`

Danger: `.alert-danger`

Warning: `.alert-warning`

Info: `.alert-info`

Light: `.alert-light`

Dark: `.alert-dark`

Here's an example of a basic Bootstrap alert:

```
<div class="alert alert-success" role="alert">
  This is a success alert.
</div>
```

You can also add a close button to the alert by including a `<button>` element with the `.btn-close` class inside the alert:

```
<div class="alert alert-danger alert-dismissible fade show" role="alert">
  <strong>Warning!</strong> This is a dangerous operation.
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-  label="Close"></button>
</div>
```

Types of Pagination:

Bootstrap 5 provides two types of pagination styles: the default pagination and the disabled pagination. Pagination is used to divide content into several pages, making it easier for users to navigate through large amounts of data.

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
  </ul>
</nav>
```

Types of Navbars:

Bootstrap 5 provides two types of navigation bars, the standard navbar and the sticky-top navbar. Navigation bars are used to provide easy navigation through a website's pages and content. The standard navbar is created using the `.navbar` class. Here's an example of a standard navbar:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">My Website</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">About Us</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Contact Us</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

The full-screen carousel is created using the `.carousel` class with the `.carousel-fade` class and the `.carousel-fullscreen` class. Here's an example of a full-screen carousel:

```
<div id="carouselExampleFade" class="carousel carousel-fade carousel-fullscreen" data-bs-
```

```
ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="image1.jpg" class="d-block w-100" alt="Image 1">
    </div>
    <div class="carousel-item">
      <img src="image2.jpg" class="d-block w-100" alt="Image 2">
    </div>
    <div class="carousel-item">
      <img src="image3.jpg" class="d-block w-100" alt="Image 3">
    </div>
  </div></div>
```

Refactor your application to use AngularJS components instead of directives, if you haven't already done so. This will make it easier to migrate your application to Angular.

Start creating a new Angular application alongside your AngularJS application, and gradually migrate your AngularJS components and services to Angular.

Test your migrated components thoroughly, and continue the migration process until your entire application has been migrated to Angular.

ACTIVITY LOG FOR THE FIFTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 06-Feb-2023	**Installing AngularJS 8**	Install Node.js version 10.9.0 & Install AngularJS 8 CLI, Creating first app in AngularJS 8	
Day - 2 07-Feb-2023	**AngularJS 8 Architecture**	Types of AngularJS 8, AngularJS 8 Components & Directives	
Day – 3 08-Feb-2023	**AngularJS 8 Data Binding**	Types of Data Binding, AngularJS 8 Forms	
Day – 4 09-Feb-2023	**Learned the introduction of Python**	What is Python? & What can python do?, Python syntax compare to other programming languages	
Day – 5 10-Feb-2023	**Python Installation**	Python Install, Python Quick Start	
Day –6 13-Feb-2023	**Python Syntax**	Python syntax, Python Indentation	

WEEKLY REPORT

WEEK – 5 From Date (06-02-2023) to Date (13-02-2023)

Installing AngularJS 8 And Install AngularJS 8 CLI:

The latest LTS (Long-Term Support) version of Node.js as of my knowledge cutoff in September 2021 is 14.17.6. If you would like to install that version, you can follow these steps:

Visit the official Node.js website: https://nodejs.org/en/

Click on the "Downloads" button on the homepage.

On the downloads page, you will see two versions: "Current" and "LTS." Click on the "LTS" tab.

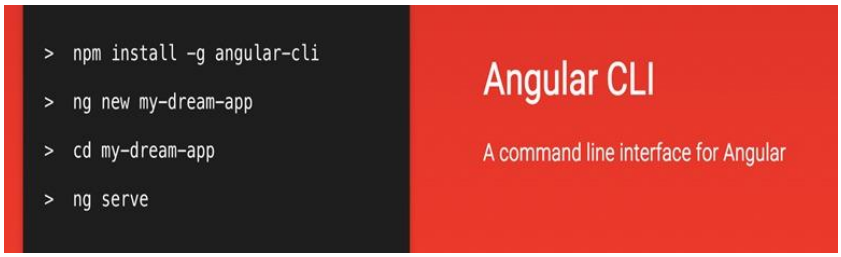
Scroll down and find the appropriate installer for your operating system. Click on the download link to start the download.

Once the download is complete, run the installer and follow the instructions provided.

After the installation is finished, you can verify that Node.js is installed by opening a terminal or **command prompt and typing the following command:**

node-v

This command will display the installed version of Node.js.



To install the Angular CLI (Command Line Interface), you can follow these steps:

You can download and install Node.js from the official website: https://nodejs.org

Open a terminal or command prompt.Install the Angular CLI globally by running the following command:npm install -g @angular/cli.This command will install the Angular CLI package globally on your system.Wait for the installation to complete. It may take a few moments.

Once the installation is finished, you can verify that the Angular CLI was installed successfully by running the following command:

ng version

This command will display the installed Angular CLI version.

Now you have successfully installed the Angular CLI on your machine. You can use it to create, develop, and manage Angular applications.

The Angular CLI provides various commands to scaffold and manage Angular projects, such as creating components, services, modules, and more. It also includes a development server, testing utilities, and build tools.

To create a new Angular project using the Angular CLI, you can run the following command:

ng new my-app

Replace "my-app" with the desired name of your project.

Creating first app in AngularJS 8:

To create your first In AngularJS Follow these steps:

Set up a basic HTML file:

Create a new HTML file and include the necessary AngularJS scripts. You can either download AngularJS from the official website (https://angularjs.org) or use a Content Delivery Network (CDN). Here's an example of a basic HTML file with AngularJS 1.8.x included:

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <title>My First AngularJS App</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
</body>
</html>
```

Define your AngularJS module:

AngularJS apps are organized into modules. Create a new module using the `angular.module` function. In this example, we'll name the module "myApp":

```
<script>
angular.module('myApp', []);
</script>
```

Add a controller:

Controllers are used to manage the behavior of a particular part of your application. Create a new controller using the `module.controller` function. Let's create a simple controller named "myController":

```
<script>
angular.module('myApp', [])
    .controller('myController', function($scope) {
        $scope.message = "Hello, AngularJS 1.8!";
    });
</script>
```

Bind the controller to your HTML:

Use the `ng-controller` directive to bind your controller to a specific element in your HTML. In this case, we'll bind it to the `<div>` element:

```
<body>
    <div ng-controller="myController">
        {{ message }}
    </div>
</body>
```

Run your app:

Open the HTML file in a web browser, and you should see the message "Hello, AngularJS 1.8!" displayed on the page. AngularJS will automatically bind the value of `$scope.message` to the `{{ message }}` expression.

Types of AngularJS 8 Architecture:

Model-View-Controller (MVC) design pattern.

Model (M): The Model represents the data and business logic of the application. In AngularJS, the model is usually created using JavaScript objects or custom JavaScript classes.

View (V): The View is the user interface that is displayed to the user. It consists of HTML templates and directives that define the structure and presentation of the application. In AngularJS, the view is defined using HTML with additional directives and bindings provided by AngularJS.

Controller (C): The Controller acts as the intermediary between the Model and the View. It handles user input, updates the model, and manipulates the view. In AngularJS, controllers are defined using the `controller` function and are associated with specific parts of the view using directives like `ng-controller`.

AngularJS 8 components and directives:

Components:

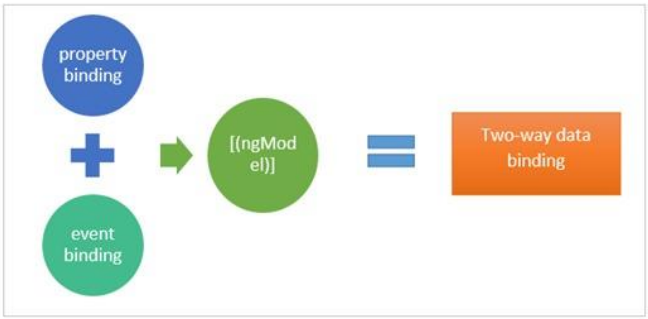
Components are reusable and self-contained units that encapsulate the behavior, view, and data of a specific part of an application. A component consists of a template (HTML markup), a controller (JavaScript code), and optionally, CSS styles. Components promote reusability and maintainability by modularizing the application into smaller, manageable pieces. They can be used to create custom elements, widgets, or entire pages in an AngularJS application.

Directives:

Directives in AngularJS are markers or attributes added to HTML elements that enhance their behavior or extend their functionality.

AngularJS 8 Data Binding

Types of Data Binding:



There are three types of data binding available:

one-way binding, two-way binding, and interpolation. Here's a brief explanation of each type:

One-Way Binding:

One-way binding allows data to flow in a single direction, from the model (data source) to the view (UI). Any changes in the model will be reflected in the view, but not vice versa. One-way binding is denoted by the `{{ expression }}` syntax. It is commonly used to display dynamic data in the view.

Two-Way Binding:

Two-way binding allows data to flow in both directions, from the model to the view and from the view to the model. Changes in either the model or the view will automatically update the other. Two-way binding is denoted by the `ng-model` directive. It is commonly used in form inputs to keep the model and view in sync.

Interpolation:

Interpolation allows you to dynamically insert values from the model into the view by embedding expressions within curly braces (`{{ }}`). It is similar to one-way binding, but it is not limited to just binding variables. Expressions can also include JavaScript function calls, arithmetic operations, and more.

AngularJS 8 Forms:

Form Element:

Start by wrapping your form elements inside an HTML `<form>` element. This allows AngularJS to track the form's state and handle form submissions.

```
<form ng-submit="submitForm()">
  <!-- Form controls go here -->
</form>
```

Input Controls:

Use AngularJS directives such as `ng-model` and `ng-required` to bind form inputs to the model and define validation requirements.

```
<input type="text" ng-model="user.name" ng-required="true">
```

Form Validation:

AngularJS provides built-in form validation directives to perform validation on form inputs. Use the `ng-required`, `ng-minlength`, `ng-maxlength`, and other validation directives to define validation rules.

Introduction of Python

What is Python?

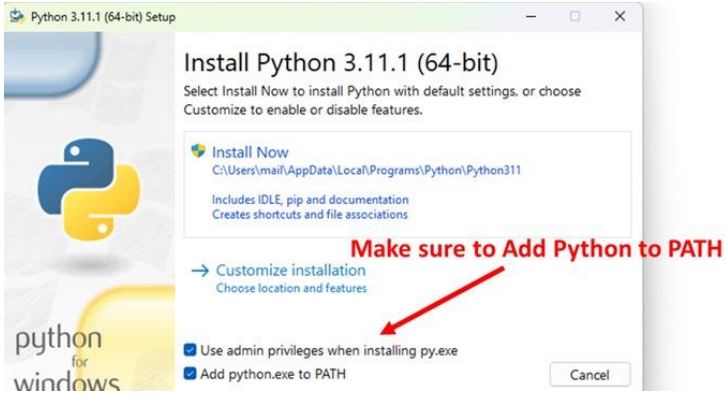
Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and uses a clean syntax, making it easy for beginners to learn and write code efficiently.

What Can Python Do?

Python is a versatile programming language that can be used to accomplish a wide range of tasks. Here are some common use cases and capabilities of Python:

Web Development: Python has several frameworks, such as Django and Flask, that facilitate web development. These frameworks provide tools and libraries for building scalable and secure web applications.

Python Installation:



The installation process for Python is straightforward and varies slightly depending on your operating system. Here's a general outline of the steps involved:

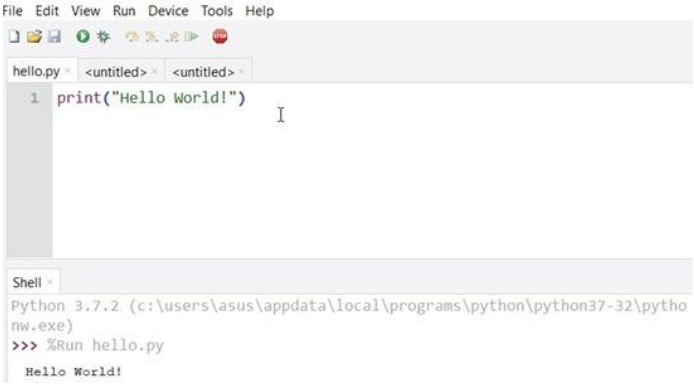
Visit the official Python website: Go to the official Python website at python.org.

Download the installer: On the website's homepage, you'll find the download section. Choose the version of Python you want to install (e.g., Python 3.9) and select the appropriate installer for your operating system (Windows, macOS, or Linux).

Run the installer: Once the installer is downloaded, run the executable file to start the installation process. Follow the on-screen instructions to proceed.

Customize the installation (optional): During the installation, you may be presented with customization options. You can choose to customize the installation location, add Python to the system's PATH

Python Quick Start:



To quickly get started with Python, follow these steps:

Install Python: Download and install the latest version of Python from the official Python website (python.org) based on your operating system. The installation process is straightforward, and you can choose the default settings.

Set up the Development Environment: Choose a code editor or integrated development environment (IDE) to write your Python code. Some popular options include Visual Studio Code, PyCharm, Atom, or Sublime Text. Install the chosen editor/IDE and ensure it has Python language support.

Write Your First Python Program: Open your code editor/IDE and create a new file with a .py extension (e.g., my_program.py). In this file, write your Python code. For example, you can start with a simple "Hello, World!" program:

print("Hello, World!")

Save and Run Your Program: Save the Python file and open a terminal or command prompt. Navigate to the directory where the file is saved. Use the following command to run your program:

python my_program.py

Python Syntax:

Python syntax is designed to be simple and readable. Here are some key aspects of Python syntax:

Indentation: Python uses indentation to define blocks of code, such as loops and conditional statements. Consistent indentation (usually four spaces) is crucial for correct code execution.

Comments: Comments in Python start with the '#' character. They are used to add explanatory notes or disable code temporarily.

This is a comment

Variables and Data Types: Variables in Python are dynamically typed, meaning you don't need to declare the variable type explicitly. Common data types include integers, floats, strings, lists, tuples, dictionaries, and booleans.

x = 5

name = "John"

my_list = [1, 2, 3]

Printing Output: The `print()` function is used to display output to the console. You can print variables, strings, or expressions by separating them with commas.

x = 10

print("The value of x is", x)

Indentation determines which statements are part of the if block and which statements are executed independently.

Proper indentation is crucial for maintaining code readability and ensuring that the code behaves as intended. It is recommended to follow the Python style guide (PEP 8) and use consistent and readable indentation practices throughout your code. Most code editors provide automatic indentation features to assist in maintaining proper indentation.

ACTIVITY LOG FOR THE SIXTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 14-Feb-2023	**Python Comments**	Creating comments, Multiple comments	
Day - 2 15-Feb-2023	**Python Variables**	Creating variables, Casting	
Day – 3 16-Feb-2023	**Python Data Types**	Built-in data types, Getting and Setting data types	
Day – 4 17-Feb-2023	**Python Numbers**	Python numbers, Type conversion	
Day – 5 20-Feb-2023	**Python Casting**	Type casting	
Day – 6 21-Feb-2023	**Python Strings**	Strings, Slicing strings	

WEEKLY REPORT

WEEK – 6 From Date (14-02-2023) to Date (21-02-2023)

Python Comments:

Creating Comments:

In Python, you can create comments to add explanatory notes or disable specific lines of code temporarily. Comments are ignored by the Python interpreter and are not executed as part of the program.

Single-line comments:

You can place a hash symbol (`#`) at the beginning of the line to create a comment. Anything after the hash symbol on that line will be treated as a comment.

Multi-line comments:

"""

This is a

multi-line comment

"""

Python Variables:

Creating variables:

In Python, variables are used to store and manipulate data. They serve as named references to values or objects. Here's how you can create variables in Python:

Variable Assignment:

variable_name = value

For example:

name = "John"

age = 25

is_student = True

Variable Naming Rules:

Variable names must start with a letter (a-z, A-Z) or an underscore (`_`).

They can contain letters, numbers, and underscores.

Variable names are case-sensitive (`name` and `Name` are different variables).

Avoid using reserved words or built-in function names as variable names.

my_variable = 10

user_name = "Alice"

age_1 = 25

```
1variable = 5  # Cannot start with a number
my-variable = 7  # Cannot contain hyphens
```

Python is dynamically typed, so you don't need to declare the variable type explicitly. The variable's type is determined based on the assigned value.

Reassigning Variables:

You can update the value of a variable by assigning it a new value.

```
count = 5
count = count + 1  # Reassigning the variable with a new value
```

Using Variables:

You can use variables in expressions, print statements, or perform operations with them.

```
x = 10
y = 5
sum = x + y
print(sum)  # Output: 15
```

Python Data Types:

Built-in data types:

Python provides several built-in data types that are fundamental for storing and manipulating different kinds of data. Here are some commonly used built-in data types in Python:

1. Numeric Types:

`int`: Represents integer values (e.g., 5, -10, 0).

`float`: Represents floating-point numbers with decimal places (e.g., 3.14, -2.5).

2. Strings:

- `str`: Represents a sequence of characters enclosed in single quotes (') or double quotes ("") (e.g., "Hello", 'Python', "12345").

3. Boolean Type:

- `bool`: Represents the truth values `True` or `False`. It is used for logical operations and control flow.

4. Lists:

- `list`: Represents an ordered collection of items enclosed in square brackets ([]). Lists can contain elements of different types and can be modified (e.g., [1, 2, "three", True]).

5. Tuples:- `tuple`: Represents an ordered collection of items enclosed in parentheses (()). Tuples are similar to lists but are immutable, meaning their elements cannot be changed once assigned (e.g., (1, 2, 3, "four")).

Python Numbers:

In Python, numbers are a fundamental data type used to represent numeric values. Python provides several built-in types for working with numbers. Here are the main types of numbers in Python:

1. Integers (`int`): Integers are whole numbers without decimal places. They can be positive or negative numbers or zero.

Example: `x = 5`

2. Floating-Point Numbers (`float`): Floating-point numbers represent numbers with decimal places. They can be positive or negative and can also be written using scientific notation.

Example: `y = 3.14`

3. Complex Numbers (`complex`): Complex numbers consist of a real part and an imaginary part. They are represented by the `complex` type and are written in the form `a + bj`, where `a` is the real part and `b` is the imaginary part.

Example: `z = 2 + 3j`

Type Conversion: In Python, type conversion (also known as typecasting or data type conversion) refers to the process of changing the data type of a value from one type to another. Python provides built-in functions that allow you to convert values between different data types.

1. **`int()`:** Converts a value to an integer data type.

   Example: `x = int(3.14)` converts the floating-point number 3.14 to an integer, resulting in `x` being assigned the value 3.

2. **`float()`:** Converts a value to a floating-point data type.

   Example: `y = float("5.6")` converts the string "5.6" to a floating-point number, resulting in `y` being assigned the value 5.6.

3. **`str()`:** Converts a value to a string data type.

   Example: `z = str(10)` converts the integer 10 to a string, resulting in `z` being assigned the value "10".

4. **`bool()`:** Converts a value to a boolean data type.

   Example: `a = bool(0)` converts the integer 0 to a boolean, resulting in `a` being assigned the value `False`.

Python Casting:

Type Casting:

In Python, type casting (also known as type conversion) refers to the process of converting a value from one data type to another. Python provides several built-in functions for type casting. Here are some commonly used type casting functions:

1. **`int()`:** Converts a value to an integer data type.

Example: `x = int(3.14)` converts the floating-point number 3.14 to an integer, resulting in `x` being assigned the value 3.

2. **`float()`:** Converts a value to a floating-point data type.

Example: `y = float("5.6")` converts the string "5.6" to a floating-point number, resulting in `y` being assigned the value 5.6.

3. **`str()`:** Converts a value to a string data type.

Example: `z = str(10)` converts the integer 10 to a string, resulting in `z` being assigned the value "10".

4. **`bool()`:** Converts a value to a boolean data type.

Example: `a = bool(0)` converts the integer 0 to a boolean, resulting in `a` being assigned the value `False`.

Python Strings:

In Python, a string is a sequence of characters enclosed in single quotes (') or double quotes (""). Strings are one of the fundamental data types and are used to represent and manipulate text data. Here are some key characteristics and operations related to strings in Python:

1. **String Creation:** Strings can be created by enclosing characters in quotes. Both single and double quotes can be used interchangeably.

Example: `my_string = 'Hello, World!'`

2. **Accessing Characters:** Individual characters in a string can be accessed using indexing. Indexing starts from 0, where 0 represents the first character.

Example: `print(my_string[0])  # Output: 'H'`

3. **String Concatenation:** Strings can be concatenated (joined together) using the `+` operator.

Example: `greeting = 'Hello'

name = 'John'

message = greeting + ', ' + name # Output: 'Hello, John'`

4. **String Length:** The length of a string can be determined using the built-in `len()` function.

Example: `length = len(my_string)  # Output: 13`

Python Slicing Strings:

In Python, slicing is a technique used to extract a portion (substring) of a string by specifying a range of indices. Slicing allows you to retrieve specific characters or sub-strings from a larger string. Here's a brief overview of slicing strings in Python:

To perform slicing, you use square brackets `[]` with the start and end indices separated by a colon `:`. The start index is inclusive, while the end index is exclusive. The syntax for slicing a string is `string[start:end]`.

ACTIVITY LOG FOR THE SEVENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 22-Feb-2023	**Python Operators**	Arithmetic, Assignment, Comparison, Logical	
Day - 2 23-Feb-2023	**Python Lists**	Lists	
Day – 3 24-Feb-2023	**Python Tuples**	Access tuples	
Day – 4 27-Feb-2023	**Python Sets**	Access set items	
Day – 5 03-Mar-2023	**Python Functions**	Creating a function	
Day –6 04-Mar-2023	**Python Lambda**	Lambda syntax	

WEEKLY REPORT

WEEK – 7 From Date (22-02-2023) to Date (04-03-2023)

Python Operators:

In Python, operators are symbols or special characters that are used to perform various operations on variables and values. They allow you to perform arithmetic, comparison, logical, and other operations in your code. Here are some commonly used operators in Python:

1. Arithmetic Operators: +, -, *, /, %, **, //
2. Comparison Operators: ==, !=, >, <, >=, <=
3. Logical Operators: and, or , not

Python Lists:

In Python, a list is a versatile data structure that allows you to store and manipulate a collection of items. It is an ordered sequence of elements enclosed in square brackets `[]` and separated by commas.

1. **Creating a List:** my_list = [1, 2, 3, 4, 5]
2. **Accessing List Elements:** Individual elements can be accessed using indexing, starting from 0.

```
print(my_list[0])  # Output: 1
print(my_list[-1])  # Output: 5
```

3. **Modifying List Elements:** List elements can be updated by assigning a new value to a specific index.

```
my_list[2] = 10
print(my_list)  # Output: [1, 2, 10, 4, 5]
```

Python Tuples:

In Python, a tuple is an ordered, immutable collection of elements enclosed in parentheses `()`. Tuples are similar to lists, but unlike lists, their elements cannot be modified once they are defined.

1. **Creating a Tuple:**

```
my_tuple = (1, 2, 3, 4, 5)
```

2. **Accessing Tuple Elements:** Individual elements can be accessed using indexing, starting from 0.

```
print(my_tuple[0])  # Output: 1
print(my_tuple[-1])  # Output: 5
```

Python Sets:

In Python, a set is an unordered collection of unique elements enclosed in curly braces `{}` or created using the `set()` function. Sets are used when you want to store a collection of distinct items and perform operations like intersection, union, and difference efficiently. Here's a brief overview of sets in Python:

1. **Creating a Set:** my_set = {1, 2, 3, 4, 5}

2. **Adding Elements to a Set:** You can add elements to a set using the `add()` method.

```
my_set.add(6)
print(my_set)  # Output: {1, 2, 3, 4, 5, 6}
```

Python Functions:

In Python, a function is a reusable block of code that performs a specific task. Functions allow you to organize your code into logical units, making it more modular and easier to understand.

1. **Defining a Function:** You can define a function using the `def` keyword, followed by the function name, parentheses `()`, and a colon `:`.

```
def greet():
 print("Hello, World!")
```

2. **Calling a Function:** To use a function, you simply need to call it by its name followed by parentheses.

```
greet()  # Output: "Hello, World!"
```

Python Lambda:

Lambda Syntax:

In Python, a lambda function is an anonymous function that can be defined in a single line. It allows you to create simple, one-line functions without explicitly naming them. Here's a brief overview of the lambda syntax in Python:

The general syntax for a lambda function is:

lambda arguments: expression

The `**lambda**` keyword is used to define the lambda function.

Arguments are the input parameters passed to the function, separated by commas.

The expression is the computation performed by the lambda function.

Example 1: Adding two numbers using a lambda function:

```
add = lambda x, y: x + y
result = add(3, 5)
print(result)  # Output: 8
```

Example 2: Squaring a number using a lambda function:

```
square = lambda x: x ** 2
result = square(4)
print(result)#Output:16
```

ACTIVITY LOG FOR THE EIGHTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 06-Mar-2023	**Python Arrays**	Arrays	
Day - 2 07-Mar-2023	**Python Classes/Objects**	Class/Objects	
Day – 3 08-Mar-2023	**Holiday**	An account of **'HOLI'** festival	
Day – 4 09-Mar-2023	**Python Inheritance**	Inheritance	
Day – 5 10-Mar-2023	**Python Iterators**	Iterators	
Day – 6 11-Mar-2023	**Python Modules**	Modules	

WEEKLY REPORT

WEEK – 8 From Date (06-02-2023) to Date (11-03-2023)

Python Arrays:

In Python, an array is a container that can hold a fixed number of elements of the same data type. Arrays are implemented using the `array` module in Python's standard library.

1. **Importing the Array Module:** Before using arrays, you need to import the `array` module.

```
import array
```

2. **Creating an Array:** You can create an array by specifying the data type and initializing it with the elements.

```
my_array = array.array('i', [1, 2, 3, 4, 5])
```

Python Classes & Objects:

In Python, a class is a blueprint for creating objects, while an object is an instance of a class. A class defines the attributes (variables) and methods (functions) that objects of that class will have.

1. **Defining a Class:** You can define a class using the `class` keyword, followed by the class name, and a colon `:`.

```
class MyClass:
# class definition
```

2. **Creating an Object (Instance):** To create an object of a class, you simply need to call the class name followed by parentheses.

```
obj = MyClass()  # Creating an object of MyClass
```

Python Inheritance:

In Python, inheritance is a mechanism that allows a class to inherit attributes and methods from another class. The class that inherits from another class is called a derived class or subclass, and the class being inherited from is called the base class or superclass. Here's a brief overview of inheritance in Python:

1. **Creating a Base Class:** You can define a base class by creating a class with its attributes and methods.

```
class Shape:
    def _init_(self, color):
        self.color = color
    def display_color(self):
        print("Color:", self.color)
```

2. **Creating a Derived Class:**

To create a derived class, you specify the base class name in parentheses after the derived class name.

```
class Circle(Shape):
    def _init_(self, color, radius):
        super()._init_(color)  # Call the base class constructor
        self.radius = radius
    def display_radius(self):
        print("Radius:", self.radius)
```

Python Iterators:

In Python, an iterator is an object that can be iterated (looped) over. It provides a way to access elements of a collection or sequence one by one without the need to know the underlying structure of the collection. Here's a brief overview of iterators in Python:

1. **Iterable Objects:** An iterable is an object that can return an iterator. Examples of iterable objects include lists, tuples, strings, and dictionaries.

```
my_list = [1, 2, 3, 4, 5]  # Iterable list
my_string = "Hello"  # Iterable string
```

2. **Iterator Objects:** An iterator is an object that implements the `_iter()` and `next()` methods. The `iter()` method returns the iterator object itself, and the `next_()` method returns the next element in the iteration.

Python Modules:

In Python, a module is a file containing Python definitions, statements, and functions that can be imported and used in other Python programs. Modules allow you to organize and reuse code, making it easier to manage large projects. Here's a brief overview of Python modules:

1. **Creating a Module:** To create a module, you simply create a Python file with a `.py` extension and define your functions, classes, or variables within it.

```
# mymodule.py
def greet(name):
    print("Hello, " + name + "!")
def add(a, b):
    return a + b
```

2. **Importing a Module:** To use a module in another Python script, you need to import it using the `import` statement.

```
import mymodule
mymodule.greet("John")  # Output: "Hello, John!"
sum_result = mymodule.add(3, 5)
print(sum_result)  # Output: 8
```

ACTIVITY LOG FOR THE NINTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 13-Mar-2023	**Python Dates**	Dates	
Day – 2 14-Mar-2023	**Python JSON**	JSON	
Day – 3 15-Mar-2023	**Python PIP**	What is PIP?	
Day – 4 16-Mar-2023	**Python String Formatting**	String Formatting	
Day – 5 17-Mar-2023	**Python File Handling**	File handling	
Day – 6 18-Mar-2023	**Downloading Python Modules Using Command Prompt**	Tkinter, Numpy, Pandas, Scikit-Learn, Matplotlib	

WEEKLY REPORT

WEEK – 9 From Date (13-03-2023) to Date (18-03-2023)

Python Dates:

In Python, working with dates and times is made convenient by the `datetime` module, which provides classes and functions for manipulating dates, times, and time intervals. Here's a brief overview of working with dates in Python:

1. **Importing the datetime Module:** Before using the `datetime` module, you need to import it.

```
import datetime
```

2. **Current Date and Time:**

```
current_datetime = datetime.datetime.now()
print(current_datetime)  # Output: Current date and time in the default format.
```

Python JSON:

In Python, JSON (JavaScript Object Notation) is a lightweight data interchange format used for storing and transmitting structured data.

1. **Importing the json Module:**

Before using the `json` module, you need to import it.

```
import json
```

2. **Encoding (Serialization):**

```
data = {"name": "John", "age": 30}
json_string = json.dumps(data)
print(json_string)  # Output: {"name": "John", "age": 30}
```

Python PIP:

In Python, pip is a package management system used to install and manage software packages written in Python. It is the standard package manager for Python and comes bundled with Python installations from version 3.4 onwards. Here's a brief overview of working with pip in Python:

1. **Installing Packages:** To install a package using pip, you can run the following command in your terminal or command prompt:

```
pip install package_name
```

Replace `package_name` with the name of the package you want to install. For example, to install the `requests` package, you would run `pip install requests`.

Python String Formatting:

In Python, string formatting allows you to create formatted strings by substituting values into placeholders within a string. Here's a brief overview of string formatting in Python:

1. **String Concatenation:** You can concatenate strings using the `+` operator.

```
name = "John"
age = 30
message = "My name is " + name + " and I am " + str(age) + " years old."
print(message)  # Output: My name is John and I am 30 years old.
```

Python File Handling:

In Python, file handling allows you to read from and write to files on your computer. Here's a brief overview of file handling in Python:

1. **Opening a File:** To open a file, you can use the `open()` function, specifying the file path and the mode (read, write, append, etc.).

```
file = open("file.txt", "r")  # Open file.txt in read mode.
```

2. **Reading from a File:** content = file.read() # Read the entire content of the file

```
print(content)
```

Downloading Python Modules Using Command Prompt:

To download Python modules using the command prompt, you can use the pip command. Here's a step-by-step guide:

1. **Open the Command Prompt:**
   - On Windows, press Win + R, type "cmd," and press Enter.
   - On macOS or Linux, open the Terminal.
2. **Check if Python and pip are installed:**
   - Type `python --version` to check if Python is installed and the version is displayed.
   - Type `pip --version` to check if pip is installed and the version is displayed.

ACTIVITY LOG FOR THE TENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 20-Mar-2023	**Jupyter Installation**	Jupyter installation process	
Day - 2 21-Mar-2023	**Learned the introduction of SVM in Machine Learning**	Support Vector Machine	
Day – 3 22-Mar-2023	**Holiday**	An account of **'UGADI'** festival	
Day – 4 23-Mar-2023	**Random Forest Algorithms in Machine Learning**	Random Forest Algorithm	
Day – 5 24-Mar-2023	**Logistic Regression in Machine Learning**	Logistic Regression	
Day – 6 25-Mar-2023	**Decision Tree Classifier in Machine Learning**	Decision Tree Classifier	

WEEKLY REPORT

WEEK – 10 From Date (20-03-2023) to Date (25-03-2023)

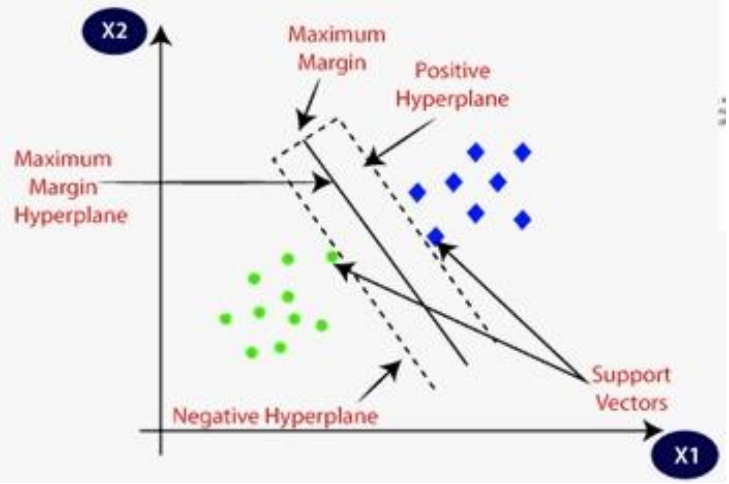
Jupyter Installation:

1. **Ensure Python is installed:** Jupyter Notebook requires Python to be installed on your system. You can download and install Python from the official Python website (https://www.python.org).

2. **Open the Command Prompt or Terminal:** pip install jupyter

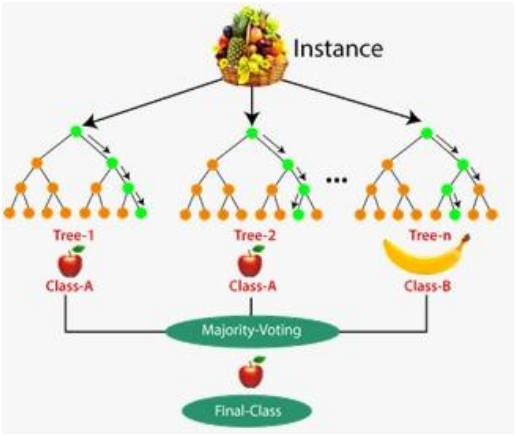
Learned the introduction of SVM in Machine Learning:

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification.




```
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X = [[2, 0], [1, 1], [2, 3], [3, 2]]
y = [0, 0, 1, 1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Random Forest Algorithm in Machine Learning:



Random Forest is a popular machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
clf = RandomForestClassifier(n_estimators=100)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:",accuracy)
```

Logistic Regression in Machine Learning:

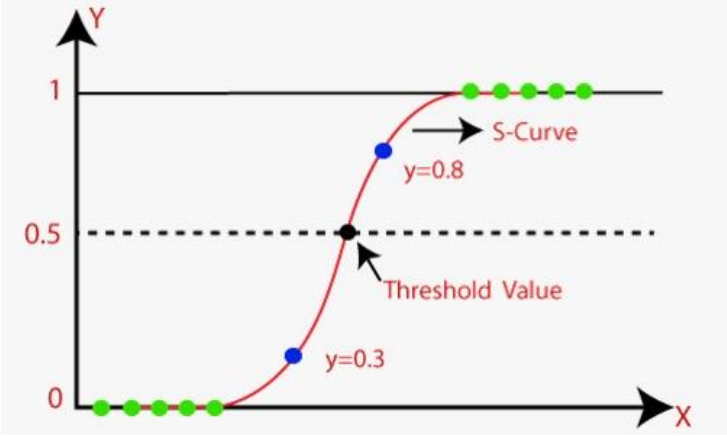
Logistic Regression is a popular machine learning algorithm used for binary classification problems. It predicts the probability of an instance belonging to a certain class based on input features.

1. Binary Classification:

Logistic Regression is mainly used for binary classification problems where the target variable has two classes, typically labeled as 0 and 1.

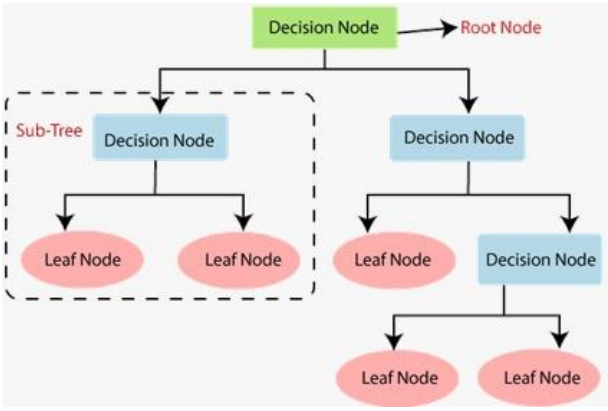
2. Logistic Function (Sigmoid):

Logistic Regression applies the logistic function, also known as the sigmoid function, to the linear combination of input features and their corresponding weights.



Decision Tree Classifier in Machine Learning:

A Decision Tree Classifier is a popular machine learning algorithm used for both classification and regression tasks. It creates a tree-like model of decisions based on input features and their corresponding target labels. Here's a brief overview of the Decision Tree Classifier:



1. **Tree Structure:**

A Decision Tree Classifier is built by recursively splitting the data based on the values of input features. The splits are made to maximize the homogeneity of target labels within each resulting subset.

2. **Feature Selection:**

The algorithm selects the best feature to split the data based on certain criteria, such as Gini impurity or information gain. These criteria measure the purity or disorder of the target labels within each subset.

ACTIVITY LOG FOR THE ELEVENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 26-Mar-2023	**Label Encoding in Machine Learning**	Label Encoding	
Day - 2 27-Mar-2023	**Normalizer in Machine Learning**	Normalizer	
Day – 3 28-Mar-2023	**One-Hot Encoding in Machine Learning**	One-Hot Encoding	
Day – 4 29-Mar-2023	**MySQL Introduction**	Basics of MySQL	
Day – 5 30-Mar-2023	**Holiday**	An account of 'SRI RAMA NAVAMI' festival	
Day –6 31-Mar-2023	**How to Install MySQL**	Installing MySQL driver, Server, Workbench	

WEEKLY REPORT

WEEK – 11 From Date (26-03-2023) to Date (31-03-2023)

Label Encoding in Machine Learning:

Label Encoding is a process used in machine learning to convert categorical variables into numerical form. It assigns a unique numerical value to each unique category or label in the variable.

1. **Categorical Variables:** Examples include "red," "green," and "blue" for a color variable or "cat," "dog," and "bird" for an animal.
2. **Numerical Representation:** Machine learning algorithms typically require numerical input, which means that categorical variables need to be converted to numerical form. Label Encoding provides a way to assign unique numerical labels to each category.

Normalizer in Machine Learning:

Normalizer is a preprocessing technique used in machine learning to scale feature vectors to have unit norm. It is particularly useful when the magnitude of features varies widely across different data points.

1. **Scaling Feature Vectors:** In many machine learning algorithms, the magnitude of feature values can have a significant impact on the model's performance.
2. **Unit Norm:** Normalizer rescales feature vectors to have a unit norm, which means that each feature vector is scaled to have a Euclidean length of 1.

One-Hot Encoding in Machine Learning: One-Hot Encoding is a popular technique used in machine learning to represent categorical variables as binary vectors.

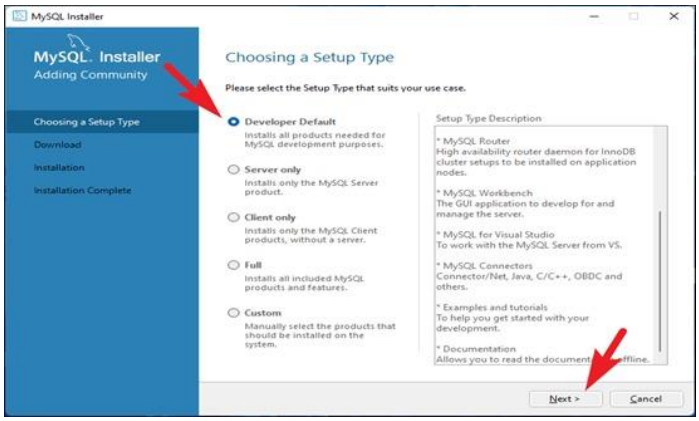
1. **Categorical Variables:** Examples include "red," "green," and "blue" for a color variable or "cat," "dog," and "bird" for an animal.
2. **Binary Representation:** It represented by a separate binary feature, where only one feature is active (1) and the others are inactive.

Introduction of MySQL

MySQL is an open-source relational database management system (RDBMS) widely used for managing and organizing large amounts of data. It provides a robust and scalable solution for storing, retrieving, and manipulating structured data. Here's a brief introduction to MySQL:

1. **Relational Database Management System (RDBMS):** MySQL is based on the relational model, which organizes data into tables consisting of rows and columns. It allows for efficient storage and retrieval of structured data using SQL (Structured Query Language).
2. **Data Management:** MySQL enables the creation, modification, and deletion of databases, tables, and related objects. It provides mechanisms for defining data types, constraints, and relationships between tables.

How to Install MySQL:



1. **Download MySQL Installer:**

Visit the official MySQL website (https://dev.mysql.com/downloads/installer/) and download the MySQL Installer appropriate for your operating system.

2. **Run the Installer:**

Locate the downloaded installer file and run it. This will start the MySQL Installer wizard

ACTIVITY LOG FOR THE TWELVETH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 01-Apr-2023	**Setup MySQL Configuration**	Setup MySQL Configuration	
Day - 2 02-Apr-2023	**How to access MySQL in Command Prompt**	MySQL Command Prompt	
Day – 3 03-Apr-2023	**Creating Table in MySQL**	MySQL table	
Day – 4 04-Apr-2023	**Learned the introduction Django Framework**	Introduction of Django	
Day – 5 05-Apr-2023	**Holiday**	An account of 'BABU JAGJIVAN RAM JAYANTI' festival	
Day –6 06-Apr-2023	**What Does Django Code Look Like?**	Django Code	

WEEKLY REPORT

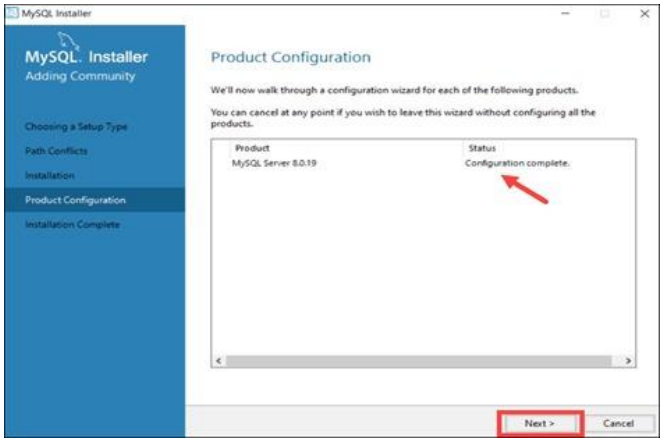
WEEK – 12 From Date (01-04-2023) to Date (06-04-2023)

Setup MySQL Configuration:

To set up MySQL configuration, you can follow these steps:

1. **Locate the Configuration File:** The MySQL configuration file is usually named `my.cnf` (Linux/macOS) or `my.ini` (Windows). Locate the file in the MySQL installation directory.

2. **Open the Configuration File:** Use a text editor to open the configuration file. Ensure you have the necessary permissions to edit the file.



How to access MySQL in Command Prompt:

1. **Open Command Prompt:** Open the Command Prompt application on your computer. On Windows, you can do this by pressing the Windows key, typing "Command Prompt," and selecting the Command Prompt application from the search results.

2. **Navigate to MySQL Bin Directory (Optional):** If the MySQL bin directory is not added to the system's PATH variable, you may need to navigate to the MySQL bin directory using the `cd` command. By default, the bin directory is located in the MySQL installation directory.

3. **Connect to MySQL:** Once you are in the Command Prompt, enter the following command to connect to MySQL:

```
mysql -u username -p
```

Creating Table in MySQL:

1. **Connect to MySQL:** Open the Command Prompt or Terminal and connect to MySQL using the `mysql` command with your username and password:

```
mysql -u username -p
```

Replace `username` with your MySQL username. Press Enter after entering the command, and you will be prompted to enter your MySQL password.

2. **Select a Database (Optional):** If you want to create the table in a specific database, use the `USE` statement to select the database. For example, to select a database named "mydatabase":

```
USE mydatabase;
    CREATE TABLE customers (
      id INT PRIMARY KEY,
      name VARCHAR(50),
      email VARCHAR(100)
    );
    SHOW TABLES;
```

Learned the introduction of Django Framework

Django is a high-level Python web framework that follows the model-view-controller (MVC) architectural pattern. It provides a robust set of tools and libraries for building web applications quickly and efficiently.

1. **Rapid Development:** Django emphasizes the principle of "Don't Repeat Yourself" (DRY), allowing developers to build web applications rapidly by providing a comprehensive set of reusable components, pre-built functionality, and automated administrative interfaces.

2. **MVC Architecture:** Django follows the MVC architectural pattern, where models define the data structure, views handle the logic and presentation of data, and templates render the HTML user interface. This separation of concerns promotes code organization and maintainability.

What Does Django Code Look Like?

Django code is written in Python and follows a specific structure and syntax. Here's an overview of what Django code typically looks like:

1. **Project Structure:** A Django project consists of multiple components, including the project directory, application directories, configuration files, and more. The project directory contains settings, URL configurations, and other project-level files.

2. **Models:** Django models represent the data structure of your application. Models are defined as Python classes that inherit from the `django.db.models.Model` base class. Each class variable represents a database field, specifying its type, constraints, and relationships with other models.

ACTIVITY LOG FOR THE THIRTEENTH WEEK

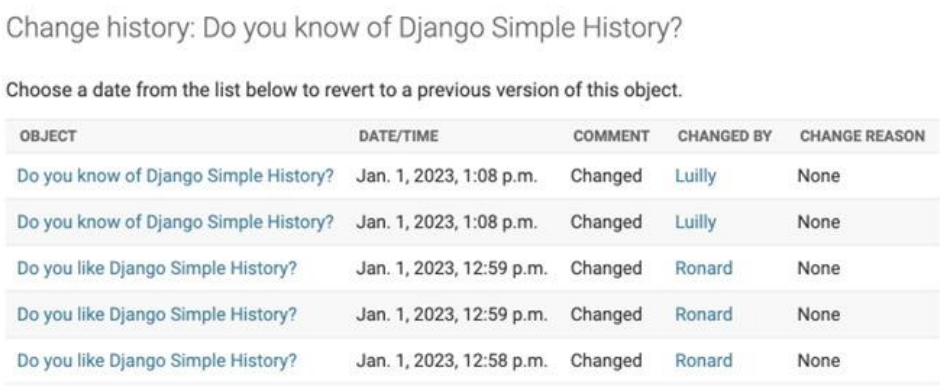
Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 07-Apr-2023	**Holiday**	An account of **'GOOD FRIDAY'** Festival	
Day - 2 08-Apr-2023	**History of Django Framework**	Django history	
Day – 3 09-Apr-2023	**Django Requires Python**	Python & Pip	
Day – 4 10-Apr-2023	**Create Virtual Environment**	Virtual Environment for Django	
Day – 5 11-Apr-2023	**Create Project or App**	Create a first Django app	
Day –6 12-Apr-2023	**Django View**	Views	

WEEKLY REPORT

WEEK – 13 From Date (07-04-2023) to Date (12-04-2023)

History of Django Framework:

Django is a popular web framework that was first released in 2005.



Change history: Do you know of Django Simple History?

Choose a date from the list below to revert to a previous version of this object.

OBJECT	DATE/TIME	COMMENT	CHANGED BY	CHANGE REASON
Do you know of Django Simple History?	Jan. 1, 2023, 1:08 p.m.	Changed	Luilly	None
Do you know of Django Simple History?	Jan. 1, 2023, 1:08 p.m.	Changed	Luilly	None
Do you like Django Simple History?	Jan. 1, 2023, 12:59 p.m.	Changed	Ronard	None
Do you like Django Simple History?	Jan. 1, 2023, 12:59 p.m.	Changed	Ronard	None
Do you like Django Simple History?	Jan. 1, 2023, 12:58 p.m.	Changed	Ronard	None

1. **Development at Lawrence Journal-World:** Django was initially developed by Adrian Holovaty and Simon Willison while working at the Lawrence Journal-World, a newspaper in Kansas, USA.

2. **Open Sourced in 2005:** Django was open-sourced and released to the public in July 2005 under the BSD (Berkeley Software Distribution) license. This allowed developers worldwide to use and contribute to the framework.

Django Requires Python and Pip: Django is a Python web framework, which means it requires Python to be installed on your system. Additionally, pip, the package installer for Python, is commonly used to install Django and its dependencies.

Python: Django is built using Python, so you need to have Python installed on your system before you can use Django. You can download the latest version of Python from the official website (python.org).

PIP: Pip is the package installer for Python, and it is usually bundled with Python installations from version 3.4 onwards. Pip allows you to easily install, upgrade, and manage Python packages.

To check if you have pip installed, open a command prompt and run: **pip --version**

Installing Django: pip install django

Create Virtual Environment for Django Framework:

1. Open a command prompt or terminal.
2. Navigate to the directory where you want to create the virtual environment.
3. Run the following command to create a new virtual environment: On Windows:

   python -m venv myenv

   On macOS/Linux:

   python3 -m venv myenv

4. Once the command completes, a new directory with the chosen name (`myenv` in this example) will be created in the current directory. This directory contains the virtual environment files and folders.

Activate the virtual environment: On Windows:

myenv\Scripts\activate

Install Django: With the virtual environment activated, you can now run: **pip install django**

Create First Django App: To create your first Django app, follow these steps:

Make sure you have Django installed in your Python environment.

Run the following command to create a new Django project:

django-admin startproject myproject

Navigate into the project directory**: 'cd myproject'**

Create your first Django app by running the following command: **python manage.py startapp myapp**

Open the `settings.py` file located in the `**myproject**` directory using a text editor. Add `'**myapp**'` to the `**INSTALLED_APPS**` list.

python manage.py runserver

Open your web browser and visit `**http://127.0.0.1:8000/`.** You should see the default Django welcome

Django Views: In Django, views are Python functions or classes that handle HTTP requests and return HTTP responses. Views are responsible for processing user requests, fetching data from the database, performing calculations, and rendering templates to generate HTML responses.

1. **Function-Based Views:** Function-based views are simple Python functions that take an HTTP request as an argument and return an HTTP response. You can define function-based views in a Django app's `**views.py**` file.

```
from django.http import HttpResponse
def my_view(request):
    return HttpResponse("Hello, World!")
```

2. **Class-Based Views:** Class-based views are Python classes that inherit from Django's built-in `View` class

```
from django.views import View
from django.http import HttpResponse
class MyView(View):
```

ACTIVITY LOG FOR THE FOURTEENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 13-04-2023	**Django Files Structure**	Files structure in Django	
Day - 2 14-04-2023	**Holiday**	An account of **'AMBEDKAR JAYANTI'** Festival	
Day – 3 15-04-2023	**Django Insert Data**	Insert Data in Django	
Day – 4 16-04-2023	**Django Delete Data**	Delete Data in Django	
Day – 5 17-04-2023	**Django Admin**	Django Admin app	
Day – 6 18-04-2023	**Create User in Django**	Creating User in Admin app using Django	

WEEKLY REPORT

WEEK – 14 From Date (13-04-2023) to Date (18-04-2023)

Django Files Structure:

In Django, the file structure follows a specific convention to organize the various components of a Django project. Here's a brief overview of the typical file structure in a Django project:

1. **Project Root Directory:** This is the main directory that contains the entire Django project.
   It usually has the same name as the project itself.
   It can also include files like `**.gitignore**`, `**README.md**`, etc.
2. **Project Configuration:** The project configuration resides in the project's root directory.
   It typically includes files like `manage.py` and the project's main settings file (`**settings.py**`).

Django Insert Data:

To insert data into a database using Django, you first need to define a model for your data. Models are Python classes that define the fields and behaviors of the data you want to store. Once you have defined your model, you can use Django's **Object-Relational Mapping** (**ORM**) to insert data into the database.

1. **First, define a model in `models.py` file:**

```
from django.db import models
class Student(models.Model):
    name = models.CharField(max_length=100)
    roll_number = models.IntegerField()
```

2. **Next, create a new instance of the model and save it to the database:**

```
from myapp.models import Student
new_student = Student(name='John Doe', roll_number=1234)
new_student.save()
```

Django Delete Data:

To delete data from a database using Django, you can use the ORM (Object-Relational Mapping) provided by Django.

1. **Retrieve the object(s) you want to delete from the database:**

```
from myapp.models import Student
students_to_delete = Student.objects.filter(roll_number=1234)
```

2. **Delete the object(s) from the database:**

```
students_to_delete.delete()
```

Django Admin:

Django provides a built-in administration interface called Django Admin, which allows you to manage your project's data and perform CRUD (Create, Read, Update, Delete) operations on your models without having to write any custom views or forms. The Django Admin is automatically generated based on your models and provides a convenient way to interact with your project's data.

1. **Create a superuser:** Run the following command in the terminal:

```
python manage.py createsuperuser
```

2. **Register your models in the admin.py file:** In your app's `admin.py` file, import your models and register them with the admin site. Here's an example of registering a `Student` model in the admin interface:

```
from django.contrib import admin
from .models import Student
admin.site.register(Student)
python manage.py runserver
```

Create User in Django:

To create a user in Django, you can use the built-in User model provided by Django's authentication system. Here's a step-by-step guide on how to create a user in Django:

1. **Import the User model:** In your **views.py** or **models.py** file, import the User model from `django.contrib.auth.models`:

```
from django.contrib.auth.models import User
```

2. **Create a new user:**

```
user = User.objects.create_user(username='example', password='password123')
user.email = 'example@example.com'
user.first_name = 'John'
user.last_name = 'Doe'
user.save()
```

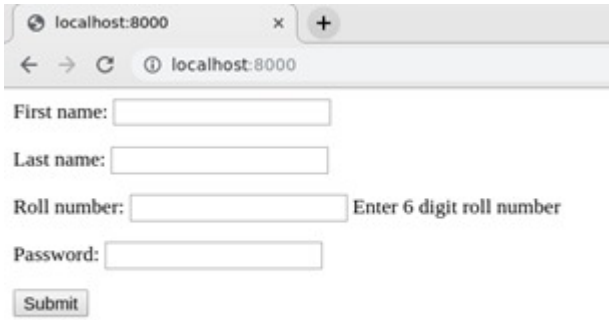
ACTIVITY LOG FOR THE FIFTEENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 19-Apr-2023	**Django Forms**	Forms in Django	
Day - 2 20-Apr-2023	**Django File Handling**	File Handling in Django	
Day – 3 21-Apr-2023	**Flask Overview**	Overview of Flask framework	
Day – 4 22-Apr-2023	**Holiday**	An account of **'Ramadan'** Festival	
Day – 5 25-Apr-2023	**Flask Application**	Application in Flask framework	
Day –6 26-Apr-2023	**Flask Routing**	Routing in Flask framework	

WEEKLY REPORT

WEEK – 15 From Date (19-04-2023) to Date (26-04-2023)

Django Forms:



In Django, forms provide a convenient way to handle user input and validate data. Django forms allow you to create HTML forms quickly and easily, and handle form submission and data validation.

```
from django import forms
class MyForm(forms.Form):
    my_field = forms.CharField(label='My Field', max_length=100)
    <form method="post">
    {% csrf_token %}
    {{ form }}
    <button type="submit">Submit</button>
</form>
```

Django File Handling:

In Django, file handling involves working with files uploaded by users or serving files stored on the server. Django provides convenient methods and features to handle file-related operations.

```
from django.db import models
class MyModel(models.Model):
   myfile = models.FileField(upload_to='uploads/')
from django import forms
class MyForm(forms.Form):
   myfile = forms.FileField()
```

Flask Overview: Flask is a popular web framework for building web applications in Python. It is lightweight, flexible, and easy to get started with. Here's a brief overview of Flask:

1. **Minimalistic:** Flask follows a "micro" framework approach, focusing on simplicity and minimalism. It provides a core set of functionalities and allows developers to add extensions as needed.

2. **Routing:** Flask uses a routing mechanism to map URLs to functions or methods that handle the corresponding requests. Routes are defined using decorators (`**@app.route**`), making it easy to define the behavior for different URLs.

Flask Application:

A Flask application is a web application built using the Flask web framework in Python. It consists of a collection of Python modules or packages that work together to handle HTTP requests, render templates, and provide the desired functionality.

```
app = Flask(_name_)
@app.route('/')
def hello():
    return "Hello, Flask!"
```

```python
if _name_ == '_main_':
    app.run()
```

(**e.g.**, **`http://localhost:5000/`).**

Flask Routing:

Flask uses a routing mechanism to map URLs to functions or methods that handle the corresponding requests. Routes are defined using decorators (`**@app.route**`), making it easy to define the behavior for different URLs.

```
from flask import Flask
app = Flask(_name_)
def hello():
    return "Hello, Flask!"
if _name_ == '_main_':
    app.run()
```

ACTIVITY LOG FOR THE SIXTEENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 27-Apr-2023	**Flask Templates**	Templates in Flask framework	
Day – 2 28-Apr-2023	**Flask Static Files**	Static files in Flask framework	
Day – 3 29-Apr-2023	**Flask Cookies**	Cookies in Flask framework	
Day – 4 01-May-2023	**Holiday**	An account of **'MAY DAY'** Festival	
Day – 5 02-May-2023	**Flask Sessions**	Sessions in Flask framework	
Day –6 03-May-2023	**Flask File Uploading**	File uploading in Flask framework	

WEEKLY REPORT

WEEK – 16 From Date (27-04-2023) to Date (03-05-2023)

Flask Templates:

In Flask, templates are used to separate the presentation logic from the application logic. Templates allow you to dynamically generate HTML pages by combining static content with dynamic data.

```
from flask import Flask, render_template
app = Flask(_name_)
@app.route('/')
def home():
    return render_template('index.html')
```

Flask Static Files:

In Flask, static files are files such as CSS stylesheets, JavaScript files, images, and other assets that are served directly to the client without any modification by the server. These files are typically stored in a directory named `static` within your Flask project.

```
<link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
<img src="{{ url_for('static', filename='img/logo.png') }}" alt="Logo">
```

Flask Cookies:

In Flask, cookies are small pieces of data stored on the client-side (browser) that are sent along with each request to the server. Cookies are often used to store user-specific information or session-related data.

Setting Cookies: from flask import Flask, make_response

```
app = Flask(_name_)
@app.route('/')
def home():
    response = make_response('Hello, Flask!')
    response.set_cookie('username', 'john')
    return response
if _name_ == '_main_':
    app.run()
```

Flask Sessions:

In Flask, sessions are a way to store user-specific data that persists across multiple requests. Sessions allow you to store information such as user authentication status, user preferences, and other session-related data.

Enabling Sessions: from flask import Flask, session

```
app = Flask(_name_)
app.secret_key = 'your_secret_key'
```

Flask File Uploading:

In Flask, file uploading allows users to upload files from their local machines to the server. This can be useful for scenarios such as uploading images, documents, or any other type of file.

1. **HTML Form:**

```
<form action="/upload" method="POST" enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit" value="Upload">
</form>
```

2. **Flask Route:** from flask import Flask, request

```
app = Flask(_name_)
@app.route('/upload', methods=['POST'])
def upload():
    file = request.files['file']
    file.save('uploads/' + file.filename)
    return 'File uploaded successfully.'
if _name_ == '_main_':
    app.run()
```

ACTIVITY LOG FOR THE SEVENTEENTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day – 1 04-May-2023	**Document Preparation - Part1**	Summary Typing	
Day - 2 05-May-2023	**Document Preparation – Part2**	Daily Report Typing	
Day – 3 06-May-2023	**Document Preparation – Part3**	Activity Report Typing	
Day – 4 08-May-2023	**Document Preparation – Part4**	Internship Outcomes	
Day – 5 09-May-2023	**Document Preparation – Part5**	Evaluation Typing	
Day –6 10-May-2023	**Document Preparation – Part6**	Invigilator Support	

WEEKLY REPORT

WEEK – 17 From Date (04-05-2023) to Date (10-05-2023)

Resources for preparing the documentation: -

HTML, CSS, JavaScript, Bootstrap	**https://www.w3schools.com/**
AngularJS 8, Typescript, Jupyter Notebook	**https://www.javatpoint.com/**
Python, Machine Learning, MySQL	**https://docs.python.org/, https://www.w3schools.com/ai/ai_machine_learning.asp**
Django, Flask	**https://pythonbasics.org/flask-vs-django/**

CONCLUSION:

As per the project timeline, week 17 is dedicated to the final stage of the project, including the printing, binding, and delivering of the final product, as well as the final presentation to the audience.

The week begins with preparing for the printing and binding of the final project. This involves ensuring that all documents and materials are in order and ready for printing. Any necessary edits or revisions should be completed, and the presentation should be reviewed to ensure it is ready for final delivery. Once everything is in order, the printing and binding process can begin.

The printing and binding process can take some time, depending on the size and complexity of the project. It is important to give ample time for this stage of the process to ensure that the final product is of high quality and meets all necessary requirements.

The final step is to deliver the project to the concerned authority. This can be done in person or via electronic means, depending on the specific requirements of the project. It is important to ensure that the final product is delivered on time and in the correct format.

After the final presentation, it is important to reflect on the overall project and presentation. This involves identifying areas of strengths and areas that need improvement, and setting goals for future presentations. Reflection can help to improve future presentations and ensure that the project meets all necessary requirements and exceeds expectations.

CHAPTER 5: OUTCOMES DESCRIPTION

Describe the work environment you have experienced (in terms of people interactions, facilities available and maintenance, clarity of job roles, protocols, procedures, processes, discipline, time management, harmonious relationships, socialization, mutual support and teamwork, motivation, space and ventilation, etc).

The work environment refers to the physical and social setting in which work is conducted. The quality of the work environment can significantly impact employee satisfaction, productivity, and well-being. In terms of people interactions, a positive work environment should promote harmonious relationships, socialization, mutual support, and teamwork among employees. Facilities available and their maintenance should be adequate and well- maintained. Clarity of job roles, protocols, procedures, and processes can help ensure employees understand their responsibilities and perform their duties efficiently. Discipline is necessary to maintain a productive work environment, and time management is essential to ensure tasks are completed on time. Ventilation and space should be considered to maintain a comfortable and safe work environment. Motivation should also be fostered to encourage employees to be engaged and productive. Overall, a positive work environment should provide employees with the necessary support and resources to
excel in their roles.

Describe the real time technical skills you have acquired (in terms of the job- related skills and hands on experience)

Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered. So what should a brand do to capture that low hanging fruit?

With the recent advances in deep learning, the ability of algorithms to analyze text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool for doing in-depth research. We believe it is important to classify incoming customer conversation about a brand based on following lines:

Key aspects of a brand's product and service that customers care about. Users' underlying intentions and reactions concerning those aspects. These basic concepts when used in combination, become a very important tool for analyzing millions of brand conversations with human level accuracy. In the post, we take the example of Uber and demonstrate how this works.

Intent Analysis

Intent analysis steps up the game by analyzing the user's intention behind a message and identifying whether it relates an opinion, news, marketing, complaint, suggestion, appreciation or query.

Contextual Semantic Search (CSS)

Now this is where things get really interesting. To derive actionable insights, it is important to understand what aspect of the brand is a user discussing about. For example: Amazon would want to segregate messages that related to: late deliveries, billing issues, promotion related queries, product reviews etc.

Describe the managerial skills you have acquired (in terms of planning, leadership, team work, behaviour, workmanship, productive use of time, weekly improvement in competencies, goal setting, decision making, performance analysis, etc.

As a student, I have acquired several managerial skills through my academic and extracurricular activities. I have developed planning skills through setting short-term and long-term academic goals and creating study schedules. I understand the importance of effective leadership, and I have developed my leadership skills by taking on leadership roles in student organizations and group projects.

Working collaboratively with other students in group projects and extracurricular activities has helped me to develop teamwork skills. I recognize the significance of behavior in achieving academic and personal success, and I have developed good work ethics and professional behaviour.

I believe in utilizing time productively, and I have developed time management skills by prioritizing tasks and creating schedules to achieve academic and personal goals. I am committed to continuous learning and have developed weekly improvement strategies to improve my competencies in various areas.

Goalsetting is an essential managerial skill, and I have learned how to set SMART (specific, measurable, attainable, relevant, and time-bound) goals for my academic and personal life. I have also learned decision-making skills by analyzing information and making sound decisions based on facts and logical reasoning.

Performance analysis is crucial for self-improvement, and I have developed skills in analyzing my academic performance to identify areas that need improvement. I have also learned how to provide constructive feedback to team members and accept feedback from others to enhance team performance.

Describe how you could improve your communication skills (in terms of improvement in oral communication, written communication, conversational abilities, confidence levels while communicating, anxiety management, understanding others, getting understood by others, extempore speech, ability to articulate the key points, closing the conversation, maintaining niceties and protocols, greeting, thanking and appreciating others, etc.,)

As a student, I recognize the importance of effective communication skills, and I am committed to improving my oral and written communication abilities. To improve my oral communication skills, I plan to practice extempore speeches and presentations to enhance my ability to articulate key points clearly and confidently. I also plan to work on managing anxiety while communicating by using techniques such as deep breathing and visualization.

To improve my written communication skills, I plan to read more extensively to improve my vocabulary and grammar skills. I also plan to practice writing and seek feedback from others to improve my writing style and clarity.

I am aware that conversational abilities play a vital role in effective communication, and I plan to work on my active listening skills to understand others better. I also plan to practice summarizing what others have said to ensure that I have understood them correctly.

Confidence is an essential aspect of effective communication, and I plan to work on building my confidence levels by practicing communication in various contexts. I will also focus on maintaining niceties and protocols, such as greeting, thanking, and appreciating others, to build positive relationships.

I understand the importance of being understood by others, and I plan to work on my clarity of expression by paying attention to my tone, pace, and pronunciation. I will also practice using simple and straightforward language to convey my thoughts and ideas effectively.

Closing a conversation is also a crucial aspect of effective communication, and I plan to work on wrapping up discussions concisely and positively. I will practice ending conversations on a positive note to build a good rapport with others.

Describe how could you could enhance your abilities in group discussions, participation in teams, contribution as a team member, leading a team/activity.

As a student, I recognize the importance of participating effectively in group discussions, contributing to teams, and leading team activities. To enhance my abilities in group discussions, I plan to improve my active listening skills and learn how to ask effective questions that encourage discussion. I also plan to research and prepare before the discussion to contribute to the conversation effectively.

To improve my participation as a team member, I plan to develop my teamwork skills, including effective communication, collaboration, and problem-solving. I will also work on being reliable and accountable by meeting deadlines and fulfilling commitments. I believe that constructive feedback is essential to improving as a team member, and I plan to seek feedback from others to identify areas for improvement.

As I develop my skills as a team member, I plan to work towards taking on leadership roles in team activities. To develop my leadership skills, I will observe and learn from effective leaders and seek opportunities to lead team activities. I will also work on my ability to motivate and inspire team members to achieve common goals.

I understand that effective team leadership requires a clear understanding of team dynamics, and I plan to develop my skills in this area. I will learn how to delegate tasks, provide constructive feedback, and resolve conflicts within the team. I also plan to work on my decision-making skills to make informed and timely decisions for the benefit of the team.

Overall, I recognize that contributing effectively as a team member and leading a team requires a combination of communication, collaboration, and leadership skills. Through practice and seeking feedback, I plan to develop and refine these skills to achieve success as a team member and leader.

Student Self Evaluation of the Short-Term Internship

Student Name: SHAIK ARSHAD WASIB

Registration No: K2001041

Term of Internship: **From:** 09/Jan/2023 **To:** 10/May/2023

Date of Evaluation:

Organization Name & Address: VXL IT SOLUTIONS, Vijayawada Rd, Beside Big-C Sriram nagar, Benz circle, Vijayawada, Andhra Pradesh 520010.

Please rate your performance in the following areas:

Rating Scale: **Letter grade of CGPA calculation to be provided**

1	Oral communication	1	2	3	4	5
2	Written communication	1	2	3	4	5
3	Proactiveness	1	2	3	4	5
4	Interaction ability with community	1	2	3	4	5
5	Positive Attitude	1	2	3	4	5
6	Self-confidence	1	2	3	4	5
7	Ability to learn	1	2	3	4	5
8	Work Plan and organization	1	2	3	4	5
9	Professionalism	1	2	3	4	5
10	Creativity	1	2	3	4	5
11	Quality of work done	1	2	3	4	5
12	Time Management	1	2	3	4	5
13	Understanding the Community	1	2	3	4	5
14	Achievement of Desired Outcomes	1	2	3	4	5
15	**OVERALL PERFORMANCE**	**1**	**2**	**3**	**4**	**5**

Date: **Signature of the Student**

Evaluation by the Supervisor of the Intern Organization

Student Name: SHAIK ARSHAD WASIB

Registration No: K2001041

Term of Internship: **From:** 09/Jan/2023 **To:** 10/May/2023

Date of Evaluation:

Organization Name & Address: VXL IT SOLUTIONS, Vijayawada Rd, Beside Big-C Sriram nagar, Benz circle, Vijayawada, Andhra Pradesh 520010.

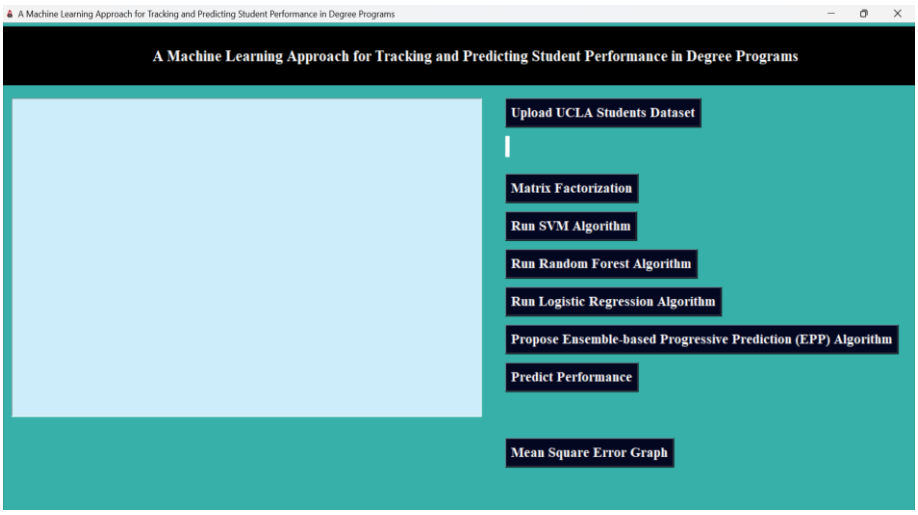
Please rate your performance in the following areas:

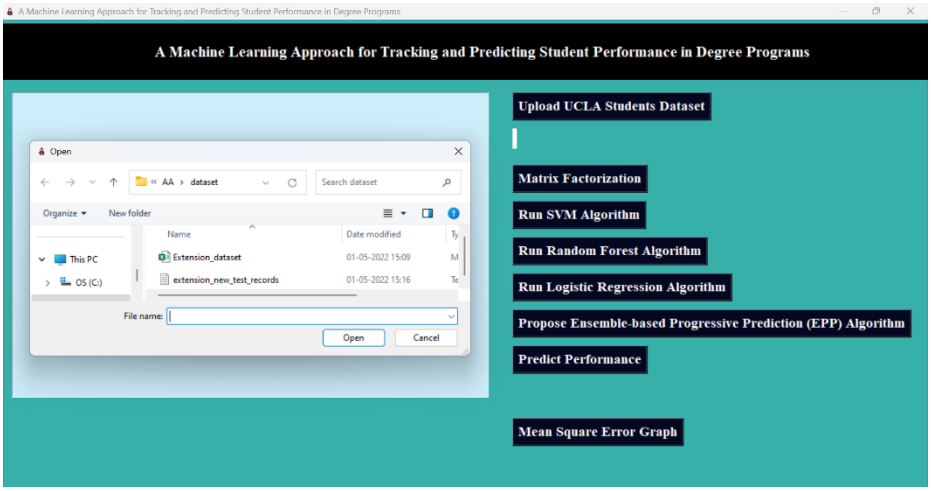
Rating Scale: **Letter grade of CGPA calculation to be provided**

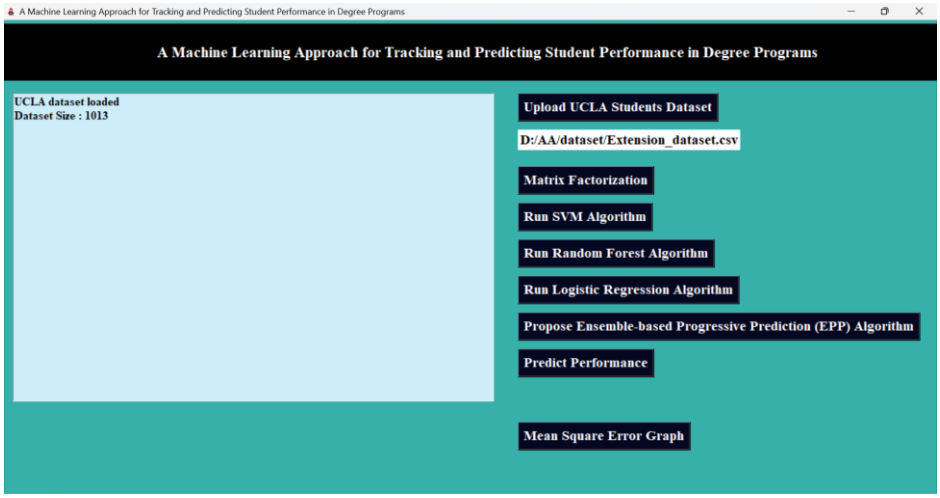
1	Oral communication	1	2	3	4	5
2	Written communication	1	2	3	4	5
3	Proactiveness	1	2	3	4	5
4	Interaction ability with community	1	2	3	4	5
5	Positive Attitude	1	2	3	4	5
6	Self-confidence	1	2	3	4	5
7	Ability to learn	1	2	3	4	5
8	Work Plan and organization	1	2	3	4	5
9	Professionalism	1	2	3	4	5
10	Creativity	1	2	3	4	5
11	Quality of work done	1	2	3	4	5
12	Time Management	1	2	3	4	5
13	Understanding the Community	1	2	3	4	5
14	Achievement of Desired Outcomes	1	2	3	4	5
15	**OVERALL PERFORMANCE**	**1**	**2**	**3**	**4**	**5**

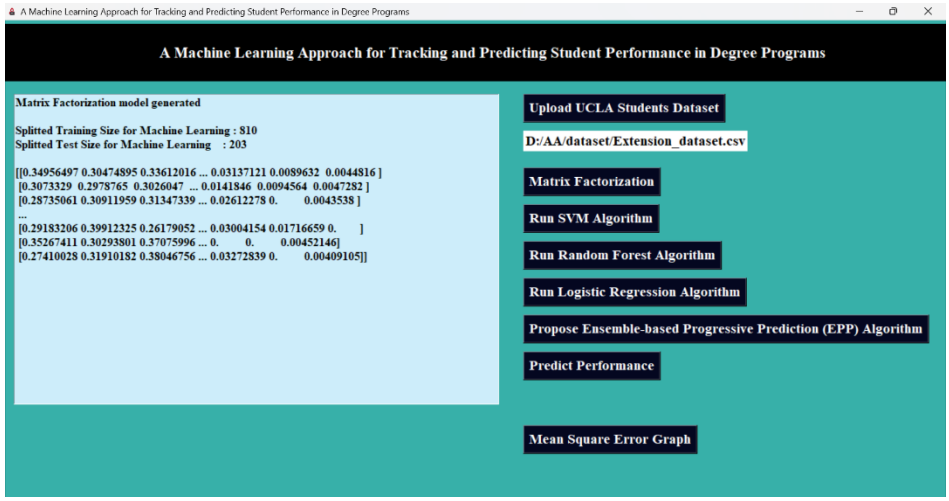
Date: **Signature of the Supervisor**

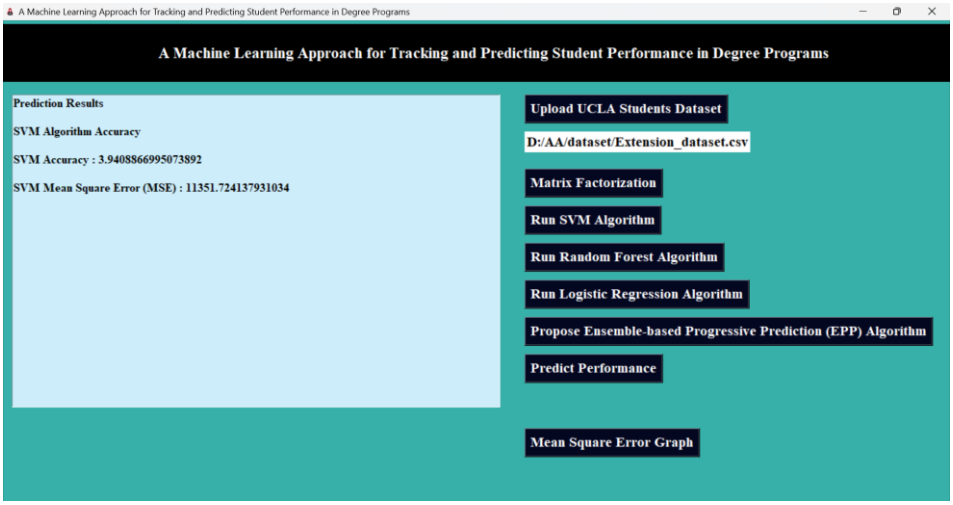
PHOTOS & VIDEO LINKS

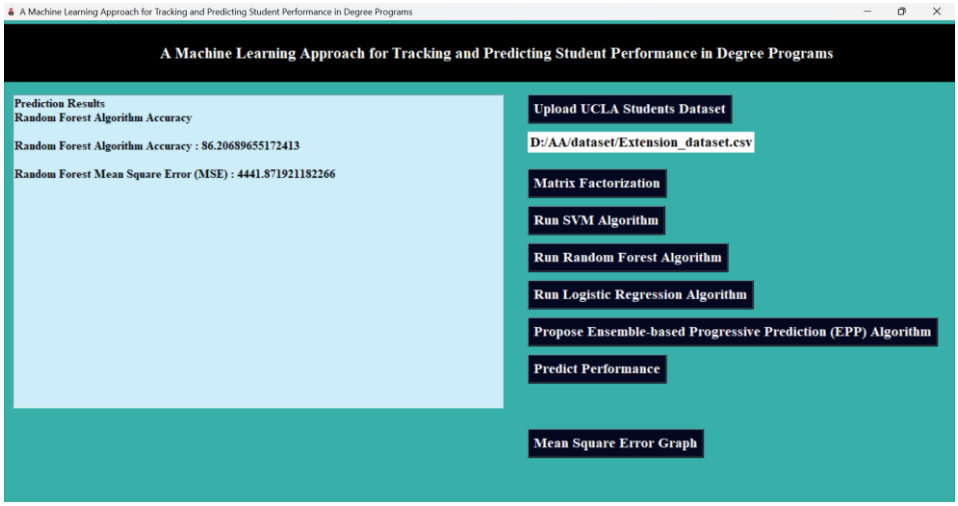


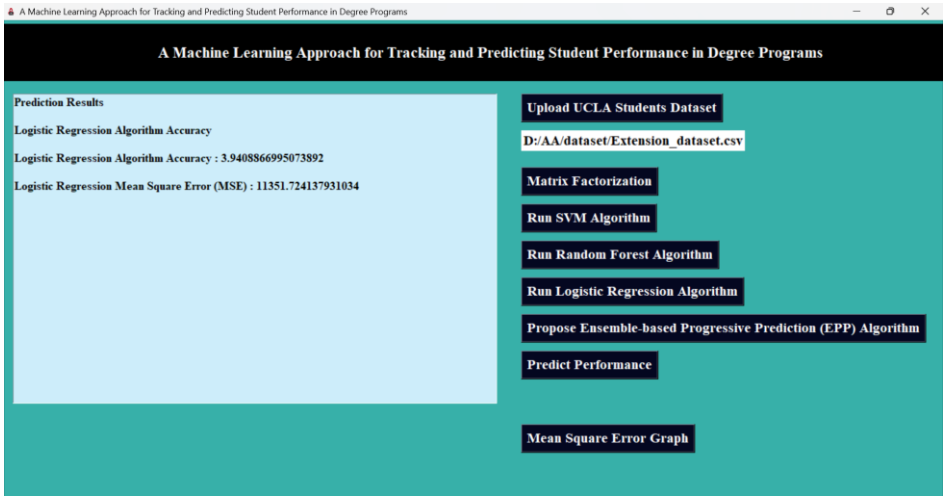


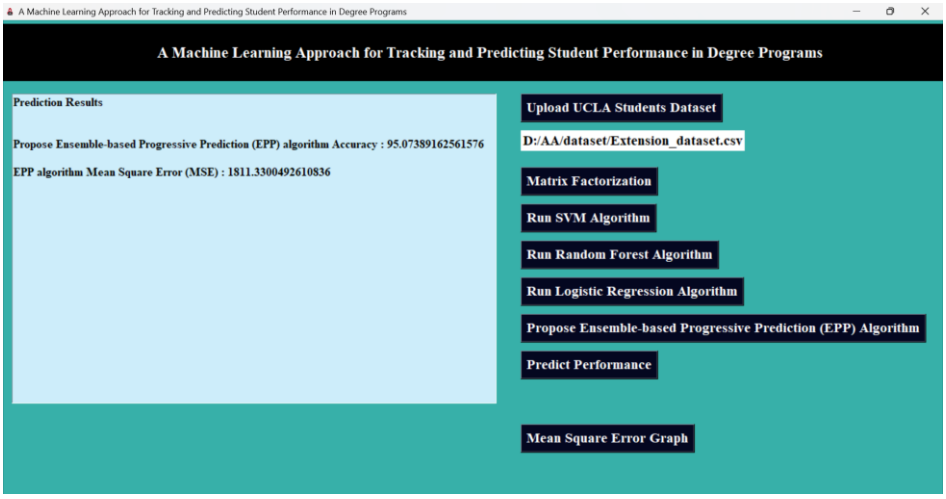


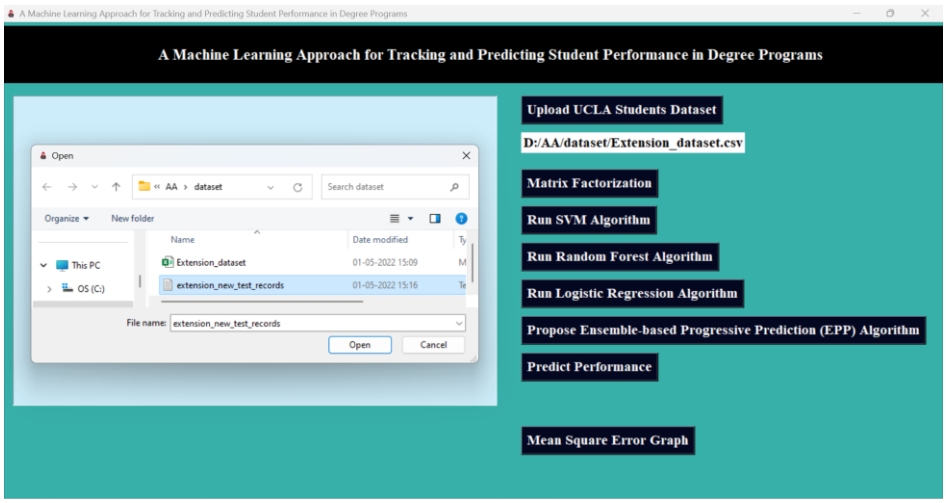

A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Matrix Factorization model generated
Splitted Training Size for Machine Learning : 810
Splitted Test Size for Machine Learning : 203
[[0.34956497 0.30474895 0.33612016 ... 0.03137121 0.0089632 0.0044816]
[0.3073329 0.2978765 0.3026047 ... 0.0141846 0.0094564 0.0047282]
[0.28735061 0.30911959 0.31347339 ... 0.02612278 0. 0.0043538]
...
[0.29183206 0.39912325 0.26179052 ... 0.03004154 0.01716659 0.]
[0.35267411 0.30293801 0.37075996 ... 0. 0. 0.00452146]
[0.27410028 0.31910182 0.38046756 ... 0.03272839 0. 0.00409105]]
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph

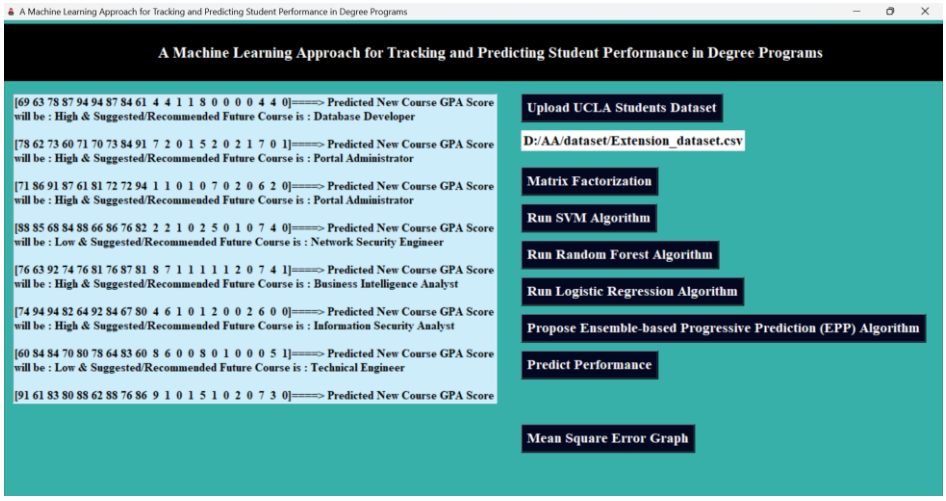

A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Prediction Results
SVM Algorithm Accuracy
SVM Accuracy : 3.9408866995073892
SVM Mean Square Error (MSE) : 11351.724137931034
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph

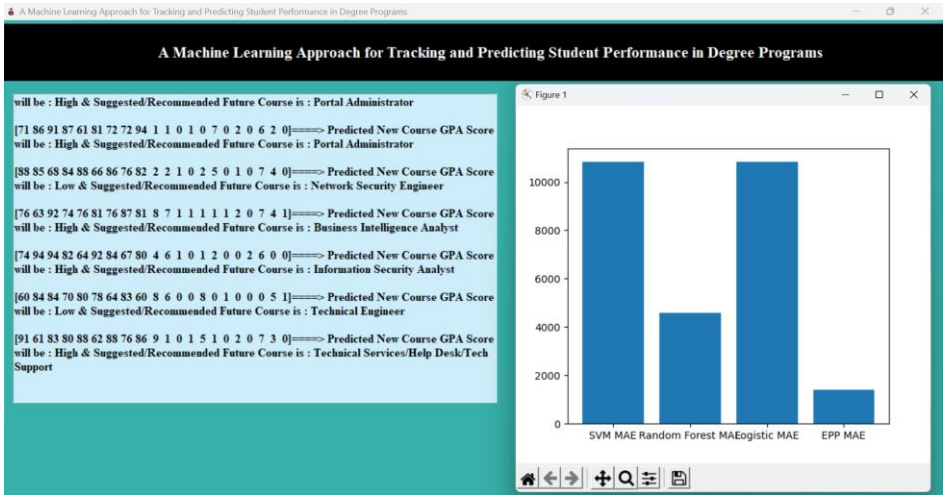

A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Prediction Results
Random Forest Algorithm Accuracy
Random Forest Algorithm Accuracy : 86.20689655172413
Random Forest Mean Square Error (MSE) : 4441.871921182266
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph


A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Prediction Results
Logistic Regression Algorithm Accuracy
Logistic Regression Algorithm Accuracy : 3.9408866995073892
Logistic Regression Mean Square Error (MSE) : 11351.724137931034
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph


A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Prediction Results
Propose Ensemble-based Progressive Prediction (EPP) algorithm Accuracy : 95.07389162561576
EPP algorithm Mean Square Error (MSE) : 1811.3300492610836
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph


A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs
Open
AA › dataset
Search dataset
Organize
New folder
Name
Date modified
Extension_dataset
01-05-2022 15:09
extension_new_test_records
01-05-2022 15:16
This PC
OS (C:)
File name: extension_new_test_records
Open
Cancel
Upload UCLA Students Dataset
D:/AA/dataset/Extension_dataset.csv
Matrix Factorization
Run SVM Algorithm
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Propose Ensemble-based Progressive Prediction (EPP) Algorithm
Predict Performance
Mean Square Error Graph





CONCLUSION:

The use of machine learning techniques in education has shown great potential in various applications, including **tracking and predicting student performance in degree programs**. By leveraging data from multiple sources, such as student demographics, academic records, and online learning platforms, machine learning algorithms can analyze patterns and make accurate predictions about student outcomes.

One of the key advantages of using a machine learning approach is its ability to handle large amounts of data and identify hidden patterns that may not be evident to human observers. By training models on historical student data, these algorithms can learn from past trends and make predictions about future performance. This information can be invaluable for educators and administrators, as it enables them to identify at-risk students early on and provide timely interventions to improve their academic success.

In conclusion, machine learning approaches offer a powerful tool for tracking and predicting student performance in degree programs. By leveraging the vast amount of data available in educational contexts, these algorithms can provide valuable insights to educators and administrators, enabling them to identify at-risk students and implement targeted interventions.

EVALUATION

Internal & External Evaluation for Semester Internship

Objectives:

- Explore career alternatives prior to graduation.
- To assess interests and abilities in the field of study.
- To develop communication, interpersonal and other critical skills in the future job.
- To acquire additional skills required for the world of work.
- To acquire employment contacts leading directly to a full-time job following graduation from college.

Assessment Model:

- There shall be both internal evaluation and external evaluation
- The Faculty Guide assigned is in-charge of the learning activities of the students and for the comprehensive and continuous assessment of the students.
- The assessment is to be conducted for 200 marks. Internal Evaluation for 50 marks and External Evaluation for 150 marks
- The number of credits assigned is 12. Later the marks shall be converted into grades and grade points to include finally in the SGPA and CGPA.
- The weightings for Internal Evaluation shall be:
    - Activity Log 10 marks
    - Internship Evaluation 30 marks
    - Oral Presentation 10 marks
- The weightings for External Evaluation shall be:
    - Internship Evaluation 100 marks
    - Viva-Voce 50 marks
- The External Evaluation shall be conducted by an Evaluation Committee comprising of the Principal, Faculty Guide, Internal Expert and External Expert nominated by the affiliating University. The Evaluation Committee shall also consider the grading given by the Supervisor of the Intern Organization.
- Activity Log is the record of the day-to-day activities. The Activity Log is assessed on an individual basis, thus allowing for individual members within groups to be assessed this way. The assessment will take into consideration the individual student's involvement in the assigned work.

- While evaluating the student's Activity Log, the following shall be considered -

  a. The individual student's effort and commitment.

  b. The originality and quality of the work produced by the individual student.

  c. The student's integration and co-operation with the work assigned.

  d. The completeness of the Activity Log.

- The Internship Evaluation shall include the following components and based on Weekly Reports and Outcomes Description

  a. Description of the Work Environment.

  b. Real Time Technical Skills acquired.

  c. Managerial Skills acquired.

  d. Improvement of Communication Skills.

  e. Team Dynamics

  f. Technological Developments recorded

MARKS STATEMENT

(To be used by the Examiners)

INTERNAL ASSESSMENT STATEMENT

Name of the Student : SHAIK ARSHAD WASIB

Program of Study : Python Full Stack Development

Year of Study : 2020 - 2023

Group : BCA

Register No/H.T.No : K2001041

Name of the College : Kakaraparti Bhavanarayana College

University : Krishna University

S. No	Evaluation Criterion	Maximum Marks	Marks Awarded
1.	Activity Log	10	
2.	Internship Evaluation	30	
3.	Oral Presentation	10	
	GRAND TOTAL	50	

Date: **Signature of the Faculty Guide**

EXTERNAL ASSESSMENT STATEMENT

Name of the Student : SHAIK ARSHAD WASIB

Program of Study : Python Full Stack Development

Year of Study : 2020 - 2023

Group : BCA

Register No/H.T.No : K2001041

Name of the College : Kakaraparti Bhavanarayana College

University : Krishna University

S. No	Evaluation Criterion	Maximum Marks	Marks Awarded
1.	Internship Evaluation	80	
2.	For the grading giving by the Supervisor of the Intern Organization	20	
3.	Viva-Voce	50	
	TOTAL	150	
GRAND TOTAL (EXT 150 M + INT.50M)		**200**	

Signature of the Faculty Guide

Signature of the Internal Expert

Signature of the External Expert

Signature of the Principal with Seal