ABSTRACT

The main objective of this website will enhance the quality of learning. Meet the learning style or needs of students. Improve the efficiency and effectiveness. Improve user-accessibility and time flexibility to engage learners in the learning process.

Our Online Learning Website that will help you to learn online and access free and paid courses from our website. we believe that everyone must be equipped with basic knowledge in Technology, as well as you use it as a medium to reach a particular goal and aim. Online-learning is an education via the Internet, network, standalone computer. Online learning is basically the network-enabled convey of skills and knowledge. Online-learning refers to using electronic applications and processes to learn. Online learning includes all forms of electronically supported learning and teaching. The information and communication systems, whether networked learning or not, serve as specific media to implement the learning process

In the 20th century, we have moved from the Industrial Age through the Information Age and now to the Knowledge Age. Knowledge and its efficient management constitute the key to success and survival for organizations in the highly dynamic and competitive world of today. We have also given a form for a free-demo courses in it which will help the student to explore our courses. If you select any course, then you will get all details of the course like course duration, course fee, course chapters. It helps students learn efficiently by gaining recourses and learning courses online. It is deemed cheaper and reliable as student can learn and practice independently without pressure and stiff competition like real classrooms.

This website helps students and also working professionals to focus on themselves and do better, student can concentrate more and make the most out of their study time. Through Online Learning Website you can learn course of your choice through online resources. Lower costs than conventional teaching approaches are provided at Online Learning Website.

S.NO	TABLE OF CONTENTS	PAGE NO

1. INTRODUCTION

1.1 OVER VIEW OF THE PROJECT

Online-learning is an education via the Internet, network. Online learning is basically the network- enabled convey of skills and knowledge. Online-learning refers to using electronic applications and processes to learn. Online learning includes all forms of electronically supported learning and teaching. The information and communication systems, whether networked learning or not, serve as specific media to implement the learning process. This often involves both out-of-classroom and in- classroom educational experiences via technology, even as advances continue in regard to devices and curriculum. Online learning is the computer and network-enabled transfer of skills and knowledge.

Online learning applications and processes include Web-based learning, computer-based learning, virtual education opportunities and digital collaboration. Content is delivered via the Internet, intranet/extranet, audio or video tape, satellite TV. That is to say Online learning systems contain both Learning Management System and Course management system. It can be self-pace or instructor-led and includes media in the form of text, image, animation, streaming video and audio. It is commonly thought that new technologies can make a big difference in education. In young ages especially, children can use the huge interactivity of new media, and develop their skills, knowledge, and perception of the world, under their parents' monitoring, of course.

Our website is an electronically supported learning system, which relies on the Internet for interaction and distribution of course material between students and working professional. Growing demand to reduce the cost of education, increasing government initiatives supporting online education, and increasing penetration of smartphones and the Internet are the factors contributingto the growth of the global online education market. Additionally, the market is also expected to be boosted owing to the increasing demand for adaptive learning. However, the availability of abundant free content and lack of awareness is limiting the market growth. Many proponents of online learning believe that everyone must be equipped with basic knowledge in Technology, as well as use it as a medium to reach a particular goal and aim. In the 20th century, we have moved from the Industrial Age through the Information Age and now to the Knowledge Age. Knowledge and its efficient management constitute the key to success and survival for organizations in the highly dynamic and competitive world of today.

1.2 AIM AND OBJECTIVE OF THE PROJECT

Our online learning website represents an innovative shift in the field of learning, providing rapid access to specific knowledge and information. It offers online instruction that can be delivered anytime and anywhere through a wide range of electronic learning solutions such as Web-based courseware, online discussion groups, live virtual classes, video and audio streaming, Web chat, online simulations, and virtual mentoring. Our website enables organizations to transcend distance and other organizational gaps by providing a cohesive virtual learning environment. Companies must educate and train vendors, employees, partners, and clients to stay competitive and it can provide such just-in-time training in a cost-effective way.

Developing and deploying effective Learning programs may require products and services supplied by a variety of vendors, leaving one to connect the dots. One way to start is to definethe goals of the desired learning solution. Definition of the goals of an Online Learning solution is driven by the following factors:

- Attracting and recruiting new students.
- Promoting our excellence in teaching and research. Providing consistent, up-to-dateinformation.
- Attracting and recruiting people to study here. Provide our best faculty to students.
- Simple, easy to use platform for student and parent engagement.
- Improved quality of learning.
- Students can also learn at their own pace.
- You can get access to many resources.

1.3 PURPOSE OF THE PROJECT

The purpose of our website is to allow people to learn for personal accomplishment or to earn a professional degree, without physically attending a traditional university or academic setting. Enhance the quality of learning and teaching. Meet the learning style or needs of students. Improve the efficiency and effectiveness. Improve user-accessibility and time flexibility toengage learners in the learning process Contribute to someone's life by sharing knowledge and experience pure joy of giving! Connect with your students from wherever you are and whenever you want. Meet the learning style or needs of students. Improve the efficiency and effectiveness. Improve user- accessibility and time flexibility to engage learners in the learning process.

The purpose of our website is to not let the pandemic affect the studies. Online learning has now become an essential role to educate students. It is now probably the only way out to continue education. The purpose of learning has always been to have a positive impact on students.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

Online course websites offer flexibility and convenience, allowing learners to access educational content from anywhere with an internet connection. However, they also come with their share of disadvantages. One significant drawback is the potential lack of personal interaction and engagement. Additionally, online courses may lack the immediacy of feedback that in-person interactions provide, potentially hindering a student's ability to grasp complex concepts. Another concern is the reliance on technology, which can be a barrier for those with limited access to reliable internet or technological resources. The absence of a structured schedule in online courses might also lead to procrastination and a lack of discipline, affecting the overall learning experience. Finally, the credibility of some online courses and the potential for cheating in a virtual environment can raise doubts about the quality and authenticity of the education provided.

2.1.1 DISADVANTAGES OF EXISTING SYSTEM

Websites which offer online courses and educational resources, have several advantages but also come with some potential disadvantages. Here are some of the main disadvantages:

Quality Control: While these platforms have course review processes, the quality of courses can vary significantly from one instructor to another. Some courses may be poorly structured, outdated, or contain inaccurate information.

Lack of Interaction: Online courses often lack the direct interaction and personal guidance that traditional classroom settings provide. This can make it more challenging for students to ask questions, receive feedback, and stay motivated.

Self-Discipline Required: Learning through online courses requires a high level of self-discipline and time management skills. Without the structure of a physical classroom and set schedules, it can be easy for students to procrastinate or fall behind.

Credibility Concerns: While many respected institutions and experts offer courses on these platforms, there is also a risk of encountering courses from unqualified or unreliable sources, which can undermine the credibility of the learning experience.

2.2 PROPOSED SYSTEM

Online course websites offer numerous advantages that make them increasingly popular for learning in today's digital age. Firstly, accessibility is a key benefit, as these platforms allow learners to access a wide range of courses from the comfort of their homes, eliminating geographical constraints. This flexibility in scheduling is another advantage, enabling individuals to learn at their own pace and fit education around their existing commitments. Additionally, online courses often provide a diverse array of subjects and specializations, catering to a broad audience with varying interests and career goals. The interactive nature of many online courses, which may include quizzes, forums, and multimedia content, enhances engagement and promotes a dynamic learning experience. Furthermore, these platforms often feature industry experts and experienced instructors, ensuring high-quality content and relevant insights. Lastly, the cost-effectiveness of online courses, compared to traditional education, makes learning more affordable and accessible to a wider demographic, contributing to the democratization of education.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

Wide Range of Courses: Online Learning offer a vast array of courses covering a wide range of topics, from programming and technology to business, marketing, personal development, and more.

Affordable Pricing: Online courses on these platforms are often more affordable compared to traditional educational programs or professional training sessions. Many courses are available at discounted prices, making them accessible to a broader audience with different budgets.

Learn from Industry Experts: Many courses are created and taught by industry professionals, subject matter experts, and experienced instructors, allowing learners to gain knowledge and insights directly from those with practical experience in their respective fields.

Self-Paced Learning: Most courses on these platforms are self-paced, allowing learners to progress through the material at their own speed, revisit concepts as needed, and fit learning into their schedules more easily.

Lifetime Access: In many cases, learners gain lifetime access to the course materials, enabling them to review the content whenever necessary or use it as a reference resource in the future.

3. SYSTEM REQUIREMENTS

System requirements are what is necessary for a client to install the web application in their system and be used hopefully without any difficulties. The aim of this is to help clients make sure they have all required tools or equipment. With analysis there are some objectives for hardware, software and any other thing that would be the minimum requirements to install the system. The minimum requirements are as follows.

3.1 Hardware Requirements

Processor : Intel Pentium 4 or equivalent, 1.6 GHz or above

RAM : 2 GB or above

Storage : 20 GB available space or more

OS : Windows 10, macOS, or any modern Linux distribution.

Display : 1024x768 or above

Hard Disc : 30GB or above

System bus : 32 bits

Internet Connection : Broadband connection for optimal performance

Browser : Latest version of browser like chrome, Firefox, Safari, Edge, etc.,

3.2 SOFTWARE REQUIREMENTS

Operation System : Windows 7+

Server-side Script : Python 3.9

Tool : Visual Studio Code

Code behind : HTML, CSS, JAVASCRIPT, BOOTSTRAP, PYTHON

Frame-work : Django

Database : MySQL

3.2.1 FRONTEND APPROACH

Front End Approach:

1. HTML
2. CSS
3. JAVASCRIPT
4. BOOTSTRAP

1. HTML - (HYPERTEXT MARKUP LANGUAGE)

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. HTML is used by the browser to manipulate text, images, and other content, to display it in the required format.

HTML elements are the building blocks of HTML pages. With HTML constructs, images, and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for textsuch as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as surround and provide information about document text and may include other tags as sub-elements.

HTML page structure: The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e., doctype declaration, html, head, title, and body elements) upon which all webpages are created.

HTML5 Advantages:

- HTML is used to build websites.
- It is supported by all browsers.

It can be integrated with other languages like CSS, JavaScript etc.,

2. CSS - (Cascading Style Sheets)

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages.

Types of CSS:

Inline CSS: Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

Internal CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e. the CSS is embedded withinthe HTML file.

External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc.). CSS property written in a separate file with .CSS extension and should be linked to t 25 HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

Advantages Of CSS:

- Create Stunning Web Site: CSS handles the look and feel part of a web page. Using CSS,you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- Become a web designer: If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.

3. JS - Javascript

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses Js provide several forms of interactivity and simplicity. Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

Features of JavaScript :

1. All popular web browsers support JavaScript as they provide built-in execution environments.

2. JavaScript follows the syntax and structure of the C programming language. Thus, it is astructured programming language.

3. JavaScript is a weakly typed language, where certain types are implicitly cast.

4. JavaScript is an object-oriented programming language that uses prototypes rather than usingclasses for inheritance.

5. It is a light-weighted and interpreted language.

It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog.

4. BOOTSTRAP

Bootstrap is a popular open-source front-end framework used for designing responsive and mobile-first websites. Developed by Twitter, it provides a set of pre-designed HTML, CSS, and JavaScript components that streamline the web development process. Bootstrap is known for its ease of use, flexibility, and the ability to create consistent and visually appealing designs across various devices and screen sizes. Here are some key features of Bootstrap,

Features of Bootstrap:

Responsive Design: One of the primary features of Bootstrap is its responsiveness. Websites built with Bootstrap automatically adjust their layout and content to provide an optimal viewing experience on different devices, such as desktops, tablets, and smartphones.

Pre-defined Components: Bootstrap comes with a variety of pre-designed UI components, such as navigation bars, buttons, forms, modals, and more. These components can be easily customized and integrated into a project, saving development time and ensuring a cohesive design.

CSS Flex and Grid support: Bootstrap utilizes modern CSS features like Flexbox and Grid to create flexible and efficient layouts. This allows developers to design complex and responsive page structures without relying solely on floats or other older layout techniques.

Extensive Documentation: Bootstrap provides comprehensive and well-organized documentation. This documentation includes examples, code snippets, and explanations for each component and feature, making it easy for developers to learn and implement Bootstrap in their projects.

Customizable styles: While Bootstrap provides a set of default styles, developers can easily customize the appearance of their websites by modifying the variables and using custom stylesheets. This allows for a unique look and feel while still benefiting from the framework's underlying structure.

3.2.2 BACKEND APPROACH

Back-End Approaches

1. PYTHON
2. DJANGO
3. MySQL

1. PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

- 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.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why do we use Python?

Python is widely used in various fields such as web development, data science, artificial intelligence, automation, and more. It is simple and readable syntax makes it accessible for beginners, while its versatility and extensive libraries make it powerful for advanced applications

- 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.

APPLICATIONS OF PYTHON

Python, a versatile and high-level programming language, finds application across various domains due to its simplicity, readability, and extensive libraries. In web development, frameworks like Django and Flask leverage Python's capabilities to build robust and scalable web applications. Python's data manipulation and analysis libraries, such as NumPy, pandas, and Matplotlib, make it a popular choice in data science and machine learning. Additionally, Python is widely used in automation and scripting tasks, thanks to its ease of integration and cross-platform compatibility. Cybersecurity experts leverage Python for tasks like penetration testing and security analysis due to its extensive libraries and frameworks. Python's application extends to game development, where libraries like Pygame facilitate the creation of 2D games. Moreover, Python is employed in the cloud computing domain, with platforms like OpenStack using it for infrastructure management.

2. DJANGO

Django is a high-level, open-source web framework for Python that follows the model-view-controller (MVC) architectural pattern. It provides a robust and efficient development environment for building web applications by emphasizing the principles of DRY (Don't Repeat Yourself) and convention over configuration. Developed to enhance productivity and simplify the complexities of web development, Django offers a variety of built-in features, including an object-relational mapping (ORM) system for database interaction, a templating engine for dynamic HTML content, and a powerful URL routing mechanism. Django also prioritizes security, incorporating built-in protection against common vulnerabilities, such as SQL injection, cross-site scripting, and cross-site request forgery.

FEATURES OF DJANGO:

High-Level Abstractions: Django provides high-level abstractions for common web development patterns, following the "Don't Repeat Yourself" (DRY) principle. This includes an Object-Relational Mapping (ORM) system that allows developers to interact with databases using Python objects, reducing the need for direct SQL queries.

Model-View-Controller (MVC) Architecture: Django follows the Model-View-Controller (MVC) architectural pattern, where models represent the data structure, views handle the presentation logic, and controllers manage user input and application flow. In Django, this is often referred to as the Model-View-Template (MVT) pattern.

Admin Interface (Django Admin): Django comes with a powerful and customizable administrative interface known as Django Admin. It is automatically generated based on the data models defined in the application, allowing developers and administrators to manage database records and content easily.

URL Routing: Django includes a flexible and expressive URL routing system that allows developers to define URL patterns and map them to corresponding views. This makes it easy to organize and maintain clean and readable URL structures.

Template Engine: Django includes a template engine that enables the separation of HTML code and Python logic. Templates support variables, control structures, and filters, making it easy to create dynamic and data-driven web pages.

3. MYSQL

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and organizing data. It falls under the category of relational databases, which means it organizes data into tables with rows and columns, and it uses Structured Query Language (SQL) for database management and manipulation.

Features of MySQL:

Open Source: MySQL is open-source software, which means its source code is freely available, and users have the freedom to modify, distribute, and enhance it according to their needs. This open nature has contributed to MySQL's popularity and widespread adoption.

Relational Database Management System (RDBMS): MySQL follows the principles of relational databases, allowing users to define and manage relationships between tables. This makes it suitable for storing and retrieving structured data.

Structured Query Language (SQL): MySQL uses SQL as its query language. SQL is a standard language for interacting with relational databases, enabling users to perform tasks such as data retrieval, insertion, updating, and deletion.

Scalability: MySQL is known for its scalability, supporting both small-scale applications and large, enterprise-level databases. It can handle high-volume transactions and is often used in web applications and content management systems.

Multi-Platform Support: MySQL is compatible with various operating systems, including Windows, Linux, macOS, and Unix. This versatility allows developers to deploy MySQL in different environments.

Community and Support: MySQL has a large and active community of users and developers. This community provides support through forums, documentation, and various online resources. Additionally, commercial support is available through MySQL Enterprise Edition.

Replication: MySQL supports replication, allowing the creation of copies (replicas) of a database on different servers. This is useful for improving performance, providing high availability, and creating backups.

Triggers and Stored Procedures: MySQL supports triggers and stored procedures, allowing developers to define custom actions and business logic within the database itself. This enhances the functionality and performance of applications.

4. SYSTEM DESIGN

4.1 DATA FLOW DESIGN

4.1.1 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) visually represents the flow of data within a system. It illustrates how data moves from external sources into the system, gets processed, and then is output to external destinations. Here's a basic Data Flow Diagram for the Online learning Web Application.

A data flow diagram is a graphical view of how data is processed in a system in terms of input and output. The Data flow diagram (DFD) contains some symbol for drawing the data flow diagram.

Level 0 DFD:

Processes:

1. User Interaction: Represents the interaction between users (customers and administrators) and the system.

2. Purchase Processing: Represents the process of processing purchase by user.

3. Course Management: Represents the process of managing courses Items by administrators.

External Entities:

1. User: Represents external users interacting with the system to browse courses, order, and manage their accounts.

2. Administrators: Represents external users responsible for managing En-roll courses, user accounts, and orders.

Data Stores:

1. Users Database: Stores user information such as names, email addresses, and passwords.

2. Database: Stores information about available purchases, including names, prices, and images.

LEVEL 0 (DFD)

Data flow diagram symbol

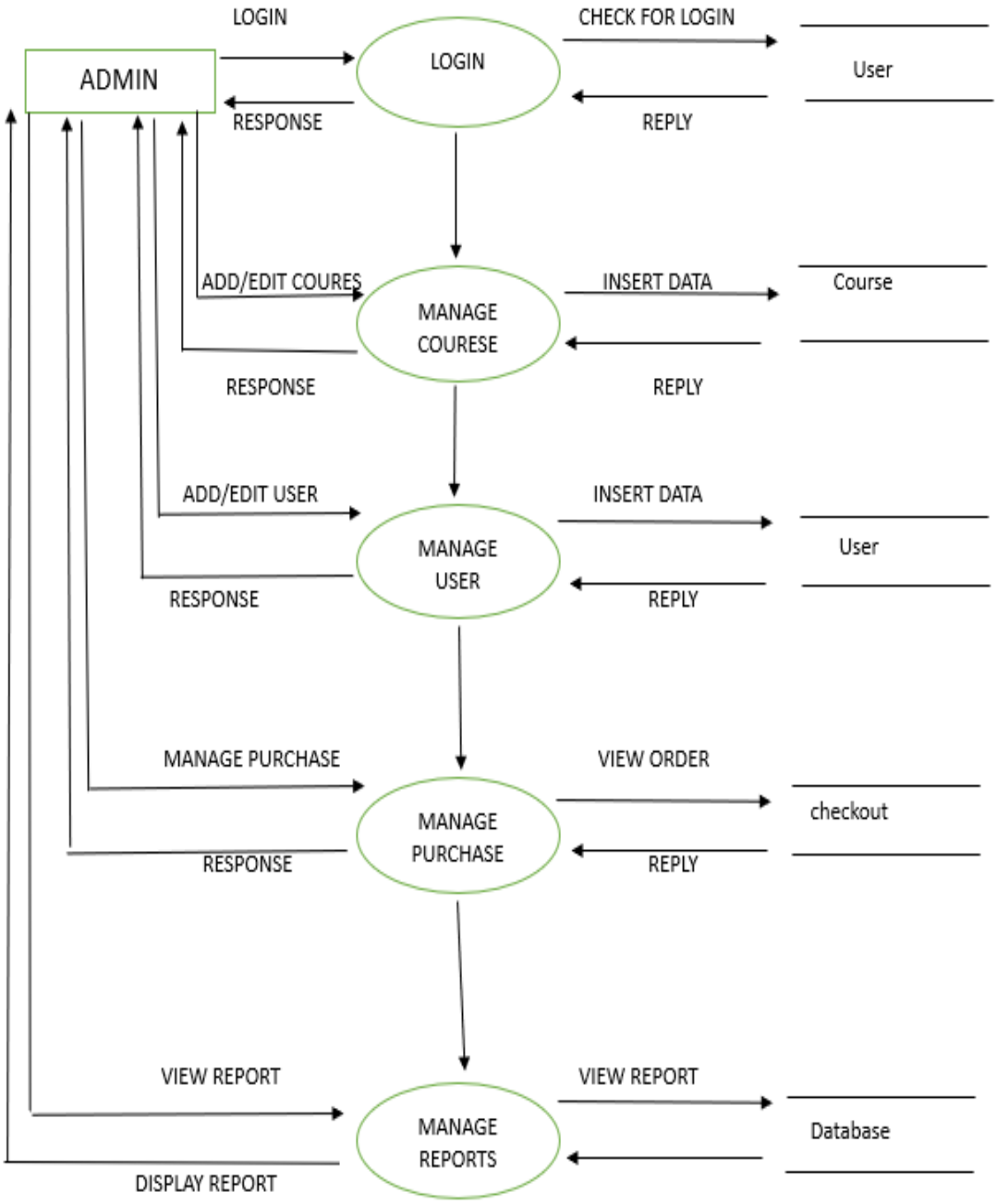
Data Flow – Data flow are pipelines through the packets of information flow.

Process : A Process or task performed by the system.

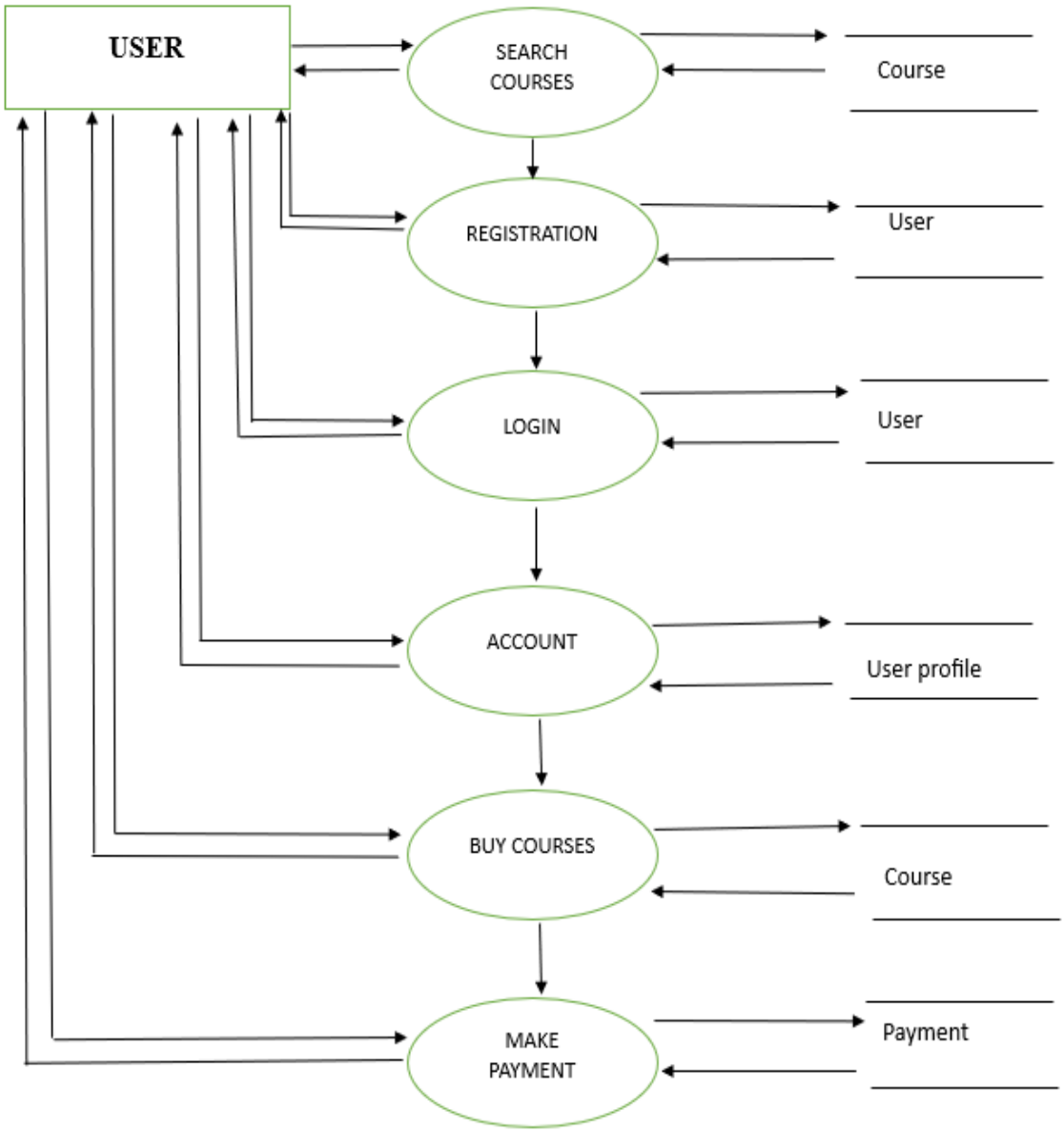
Entity : Entity are object of the system. A source or destination data of a system.

Data Store : A place where data to be stored.

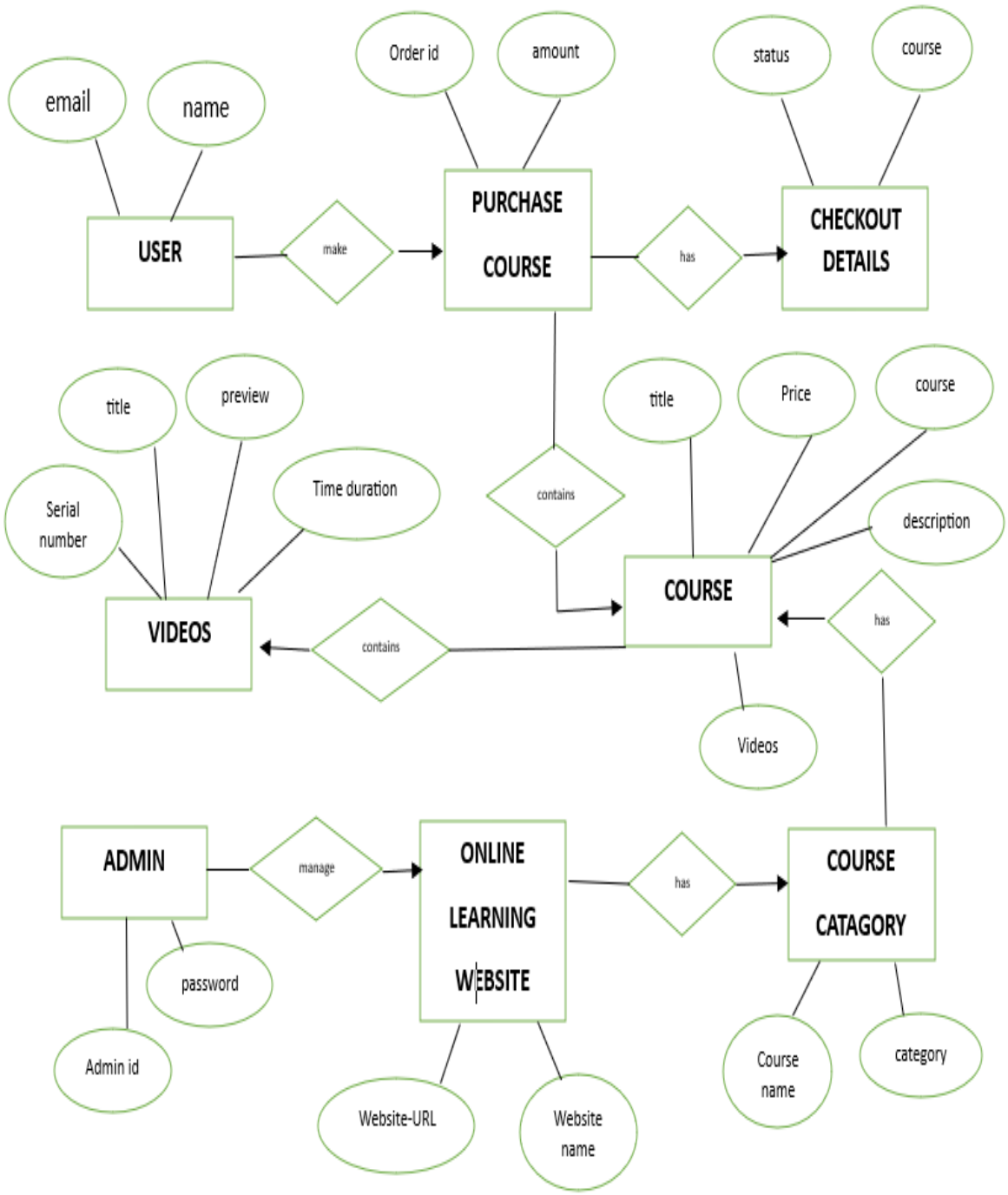
Level 1 (DFD) ADMI SIDE



Level 1 (DFD) USER SIDE



4.1.2 ER DIAGRAM



4.2 TABLE DESIGN

USER TABLE:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
User	name	CharField	primarykey
	first_name	CharField	Max length= 30
	last_name	CharField	Max length=30
	email	CharField	Max length=50

AUTHOR TABLE:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
Author	author_profile	ImageField	(AutoField)
	name	CharField	Max length=50
	about_author	TextField	Max length=500

COURSE TABLE:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
Course	featured_images	IntegerField	(AutoField)
	featured_video	CharField	Max length=300
	title	CharField	Max length=100
	created_at	datetimefield	(autofield)
	author	CharField	ForeignKey
	category	CharField	ForeignKey
	level	CharField	ForeignKey

	description	CharField	Max length=500
	price	IntegerField	(AutoField)
	discount	IntegerField	(AutoField)
	language	CharField	ForeignKey
	deadline	CharField	Max length=100
	certificate	CharField	Max length=100

VIDEO TABLE:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
Video	Serial_number	IntegerField	(AutoField)
	thumbnail	ImageField	(AutoField)
	course	CharField	ForeignKey
	lesson	CharField	ForeignKey
	title	CharField	Max length=100
	yvideo_id	IntegerField	(AutoField)
	Time_duration	IntegerField	(AutoField)
	preview	BooleanField	True or False

PAYMENT TABLE:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
Payment	Order_id	CharField	Max length=300
	Payment	CharField	Max length=300
	User	CharField	ForeignKey

	User_course	CharField	ForeignKey
	Course	CharField	ForeignKey
	Date	DateTimeField	(AutoField)
	status	BooleanField	True or False

OTHER TABLES:

TABLE NAME	FIELDS	DATATYPES	CONSTRAINTS
Categories	icon	CharField	Max Length=200
	name	CharField	Max Length=50
Level	name	CharField	Max Length=50
Language	language	CharField	Max Length=10
What_U_Learn	Course	CharField	ForeignKey
	points	CharField	Max Length=1000
Requirements	Course	CharField	ForeignKey
	points	CharField	Max Length=1000
Lesson	Course	CharField	ForeignKey
	points	CharField	Max Length=1000

4.3 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer- based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. Input design is a part of overall design, which requires careful attribute. Inaccurate input data are the most common cause of errors in data processing. The goal of designing input data is to make data entry as easy, logical and free from errors. In the system design phase input data are collected and organized in to groups of similar data.

1. User Registration – Input Form

   Fields: First Name, Last Name, Email.

   Submit Button: Register .

   Validation: Ensure valid email format, strong password.

2. User Login – Input Form

   Fields: Email, Password

   Submit Button: Login

   Validation: Ensure valid email format.

3. Purchase Courses –

   Action: click “En-Roll now” button to purchase the course.

   Confirmation: Enter Payment details to confirm Purchase.

4. Place order – Input Form

   Fields: Email, Password , First Name, Last Name.

   Submit Button: Place order.

   Validation: Ensure valid email format.

4.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. Computer output is the most important and direct source of information to the user. Output design is very important phase because the output will be in an interactive manner. The output will be in such that the user can see it from the screen and can take a hard copy from the printer. Efficient, Major form of the output is a hard copy from the printer**.**

1. Displaying Courses:

   List all available and purchased courses with details (name, price, image).

2. User Registration/Login Success:

   Display a success message upon successful registration or login.

3**.** Displaying Cart:

   List all courses added to the cart with details (name, price, quantity).

4. Displaying User Profile:

   Show user profile details including name, email, and address.

5. Logging Out:

   Provide a message confirming logout.

6. Updating Order Items/Admin Actions:

   Display success message upon adding, updating, or deleting course Items (for admin)

7. Order Confirmation:

   Display course details including total bill.

   Provide confirmation message upon successful purchase.

5. TESTING AND IMPLEMENTATION

The proposed system was developed using python. The existing system caused long time transmission process but the system developed now has a very good user- friendly tool, which has a menu-based interface, graphical interface for the end user. After coding and testing, the project is to be installed on the necessary system. The executable file is to be created and loaded in the system. Then the code is tested in the installed system. Installing the developed code in system in the form of executable file is implementation.

5.1 TYPES OF TESTING

1. Unit Testing:

Unit testing involves evaluating individual units or components of a program in isolation. Developers write test cases for functions, methods, or classes to ensure they produce the expected output for various inputs. Python provides built-in support for unit testing through the **`unitest`** module, and there are also third-party libraries like **`pytest`** and **`nose2`** that offer more features and flexibility.

2. Integration Testing:

Integration testing focuses on verifying that different units or modules of a system work together seamlessly. This type of testing ensures that the integrated components function as intended and catch issues related to their interactions. Similar to unit testing, Python developers often use frameworks like **`unittest`**,` **pytest**`, or `nose2` for integration testing.

3. Functional Testing:

Functional testing aims to validate that the software meets specified functional requirements. Test cases are designed to cover various features and user scenarios, ensuring that the application behaves correctly based on its intended functionality. Tools like `pytest`, selenium, and Behave are commonly used for functional testing in Python.

4. End-to-End (E2E) Testing:

E2E testing evaluates the entire application from start to finish, simulating real user interactions. It ensures that all components work together as expected, providing a comprehensive assessment of the system's functionality. Selenium and Cypress are popular tools for implementing E2E tests in Python.

5. Regression Testing:

Regression testing involves re-running previously executed test cases to verify that new changes or additions to the codebase do not introduce defects or negatively impact existing functionality. Python testing frameworks like `**pytest`,** `**nose2**`, and `**unittest**` are commonly used for regression testing.

6. Performance Testing:

Performance testing assesses how well a system performs under different conditions, such as varying loads or stress. Python developers can use tools like locust.io and pytest-benchmark to measure and analyze the performance characteristics of their applications.

7. Load Testing:

Load testing evaluates an application's performance under high traffic or load conditions. It helps identify performance bottlenecks and ensures the application can handle the expected number of users. Tools like locust.io and Apache JMeter can be employed for load testing in Python.

8. Security Testing:

Security testing is essential for identifying and addressing potential vulnerabilities in a software application. Python developers use tools like `OWASP` `ZAP` for dynamic analysis, and bandit for static analysis, to detect security issues in the code. Additionally, safety is used for dependency scanning to identify insecure dependencies.

5.2 SYSTEM IMPLEMENTATION

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). The system elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing; or the software realization processes of coding and testing; or the operational procedures development processes for operator's roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

6. CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only in HTML, CSS and JavaScript web-based application and no some extent Windows Application and SQL Server, but also about backend language Python. It also provides knowledge about the latest technology used in developing web enabled application. This will provide better opportunities and guidance in future in developing projects independently.

The development and implementation of our online learning platform represent a significant milestone in the realm of education and professional development. Our platform, is designed to empower learners worldwide, offering a diverse range of courses delivered by expert instructors. The project's success hinges on a robust and user-friendly interface, innovative features, and a commitment to providing high-quality, accessible education. As we navigate the dynamic landscape of online learning, we remain dedicated to continuous improvement, responsive to user feedback, and proactive in incorporating emerging technologies. Our vision is not merely a website but a catalyst for knowledge dissemination, fostering a global community of learners, and contributing to the democratization of education. With this online learning platform, we aim to inspire, equip, and connect individuals on their learning journey, bridging gaps and opening doors to a world of opportunities.

Finally, I would like to express myself as to how I find this process of developing a system to be very awaking to the mind of a student and to learn how to and teach themselves things. I have built a skill of how to search for things and develop then to my needs. It has indeed been a great experience.

7. SCOPE FOR FUTURE ENHANCEMENT

The scope for future enhancement of our project is extensive, with opportunities to enrich user experience, improve functionality, and adapt to emerging educational trends. One significant avenue for development involves the integration of advanced technologies such as virtual and augmented reality to create immersive learning environments, offering students a more engaging and interactive experience. Additionally, incorporating machine learning algorithms can enhance personalized learning paths, tailoring content to individual needs and optimizing the learning journey.

Collaboration tools and social learning features could be expanded to foster a sense of community among students, facilitating peer-to-peer interaction and group activities. Moreover, continuous updates in content delivery methods, such as incorporating microlearning modules or gamification elements, can enhance the overall effectiveness of the platform. As the educational landscape evolves, staying adaptable to new pedagogical approaches and industry demands will be crucial, ensuring the online learning website remains a cutting-edge and valuable resource for learners worldwide. Regular feedback mechanisms and data analytics can further be employed to gauge user satisfaction, track learning outcomes, and refine the platform continuously. Ultimately, the scope for future enhancement our project is dynamic, offering opportunities to embrace technological innovations and pedagogical advancements, thus staying relevant and effective in an ever-changing educational landscape.

BIBLIOGRAPHY

Books:

1. " Web Development" by Miguel Grinberg
2. "Django for Beginners" by William S. Vincent
3. "JavaScript: The Good Parts" by Douglas Crockford

Online resources:

1. Mozilla Developer Network (MDN) (https://developer.mozilla.org/en-US/)
2. Python.org(https://www.python.org/)
3. Django Documentation(https://docs.djangoproject.com/en/5.0/)

Project-Specific Resources:

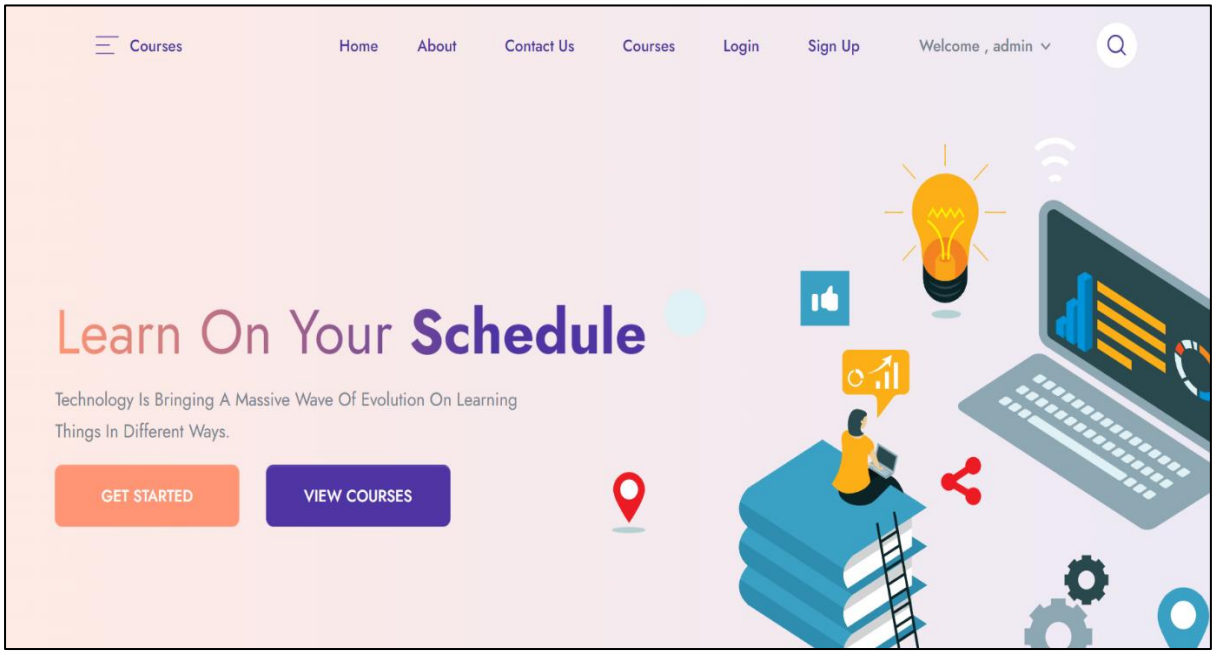
1. GitHub (https://github.com/)
2. Stack Overflow(https://stackoverflow.com/)
3. You-tube(https://www.youtube.com/@CodingEx)

By combining books, online courses, and free websites, this bibliography aims to provide a well-rounded set of resources for your online learning website project based on Python, Django, and frontend development. Additionally, incorporating project-specific resources like GitHub and Stack Overflow will enhance your ability to find practical solutions and collaborate with the development community.

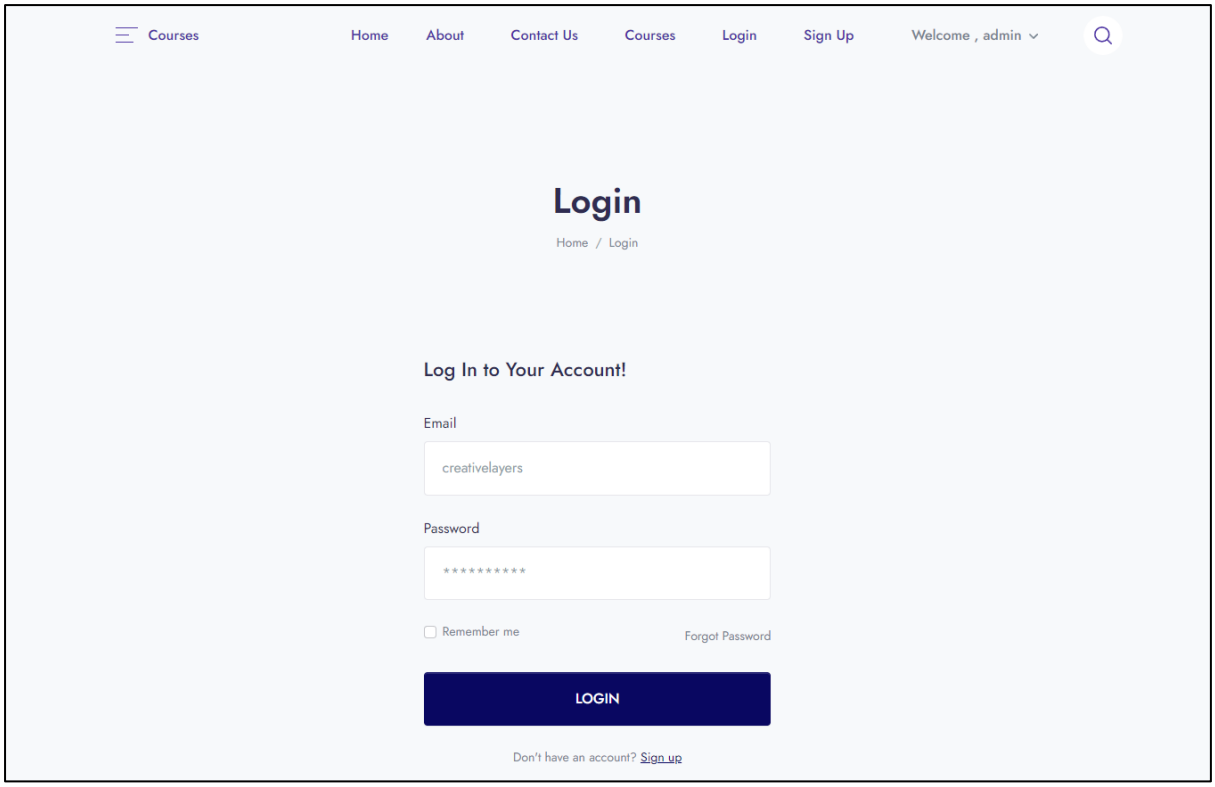
APPENDIX

A. SCREENSHOTS

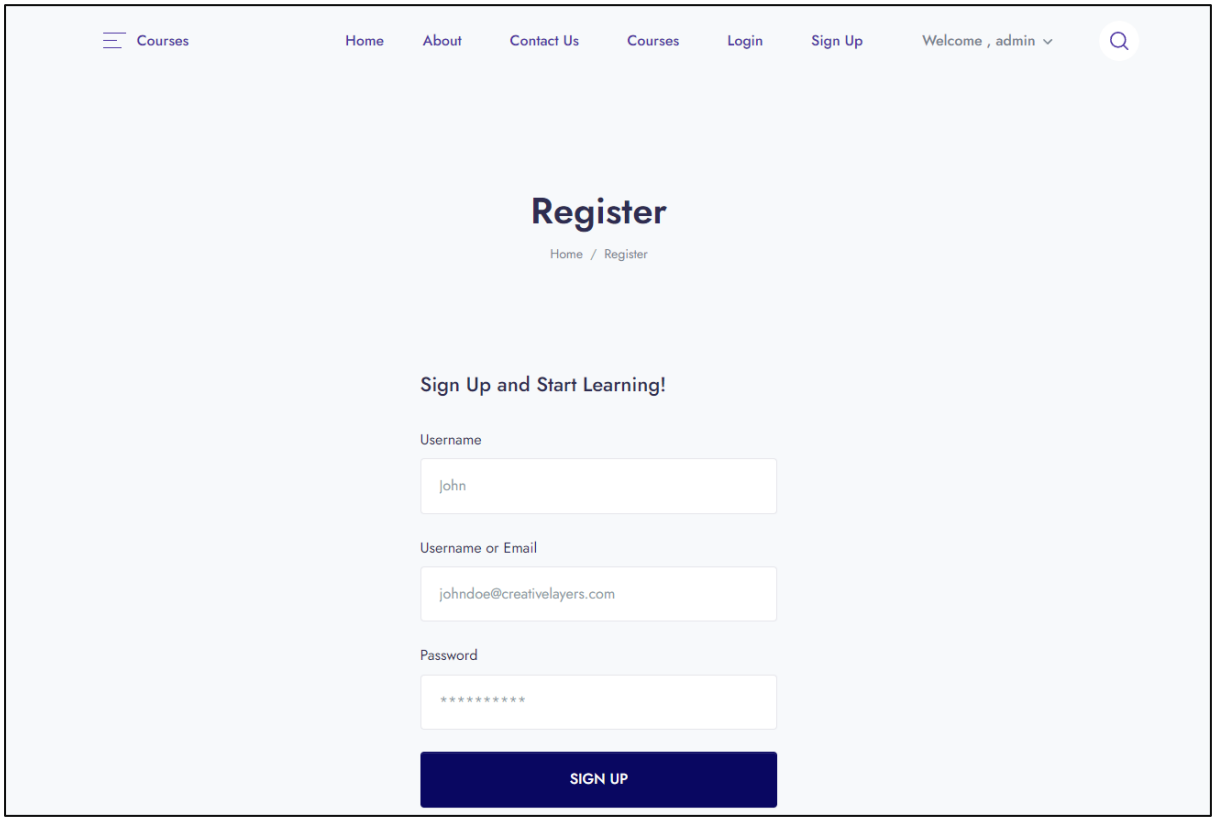
HOME PAGE:



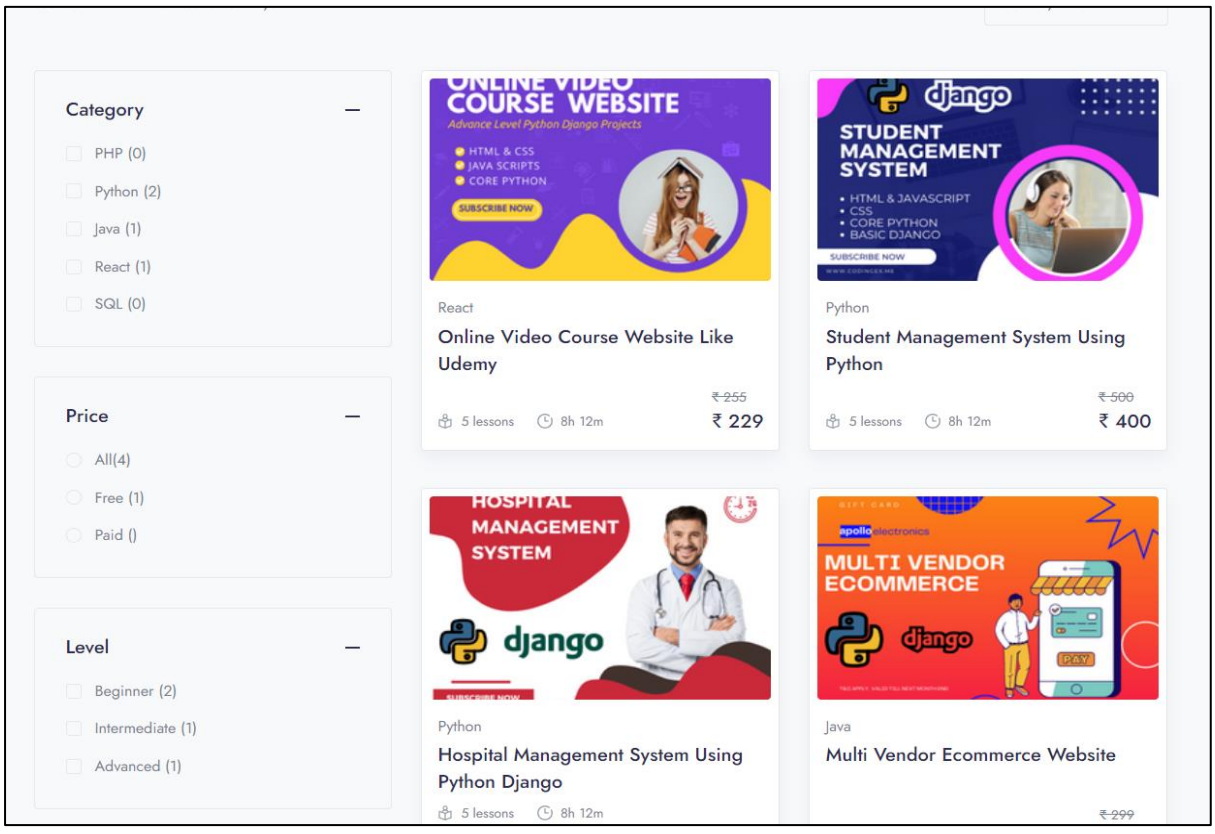
LOGIN PAGE:



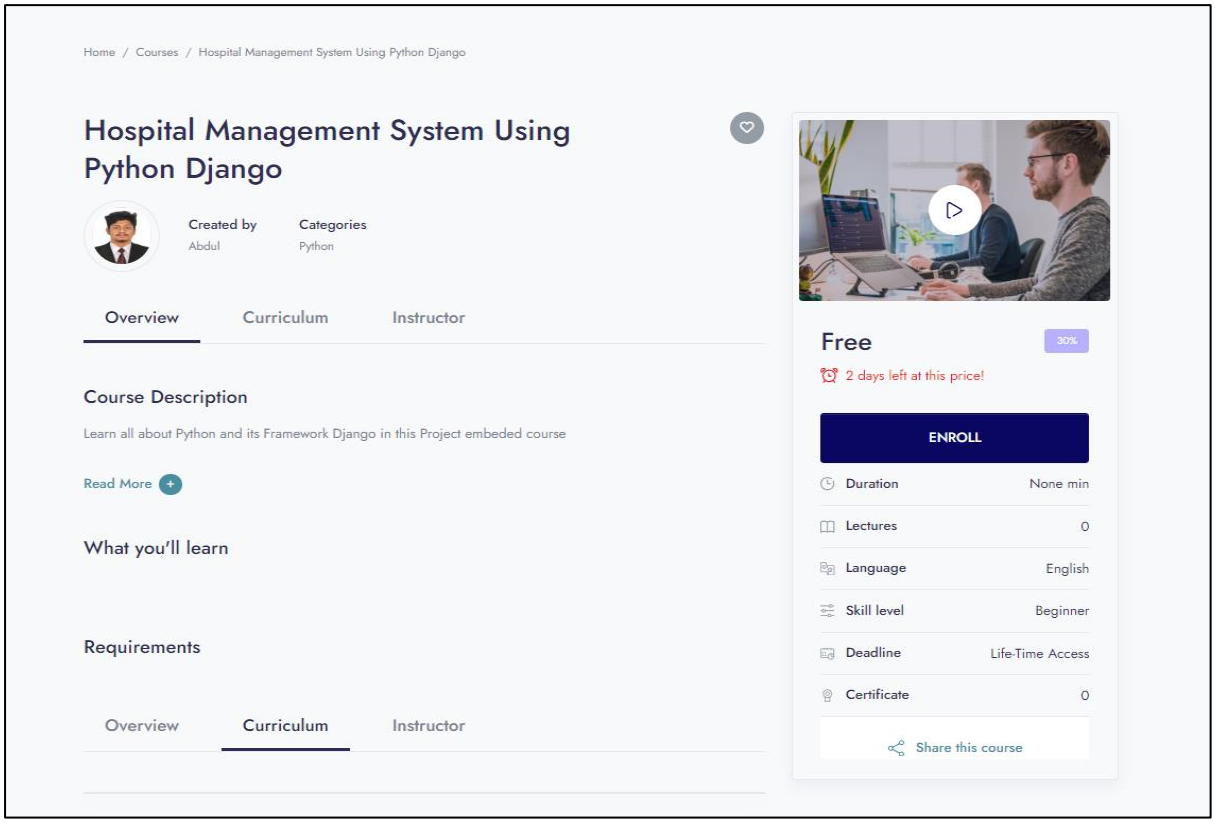
SIGNUP PAGE:



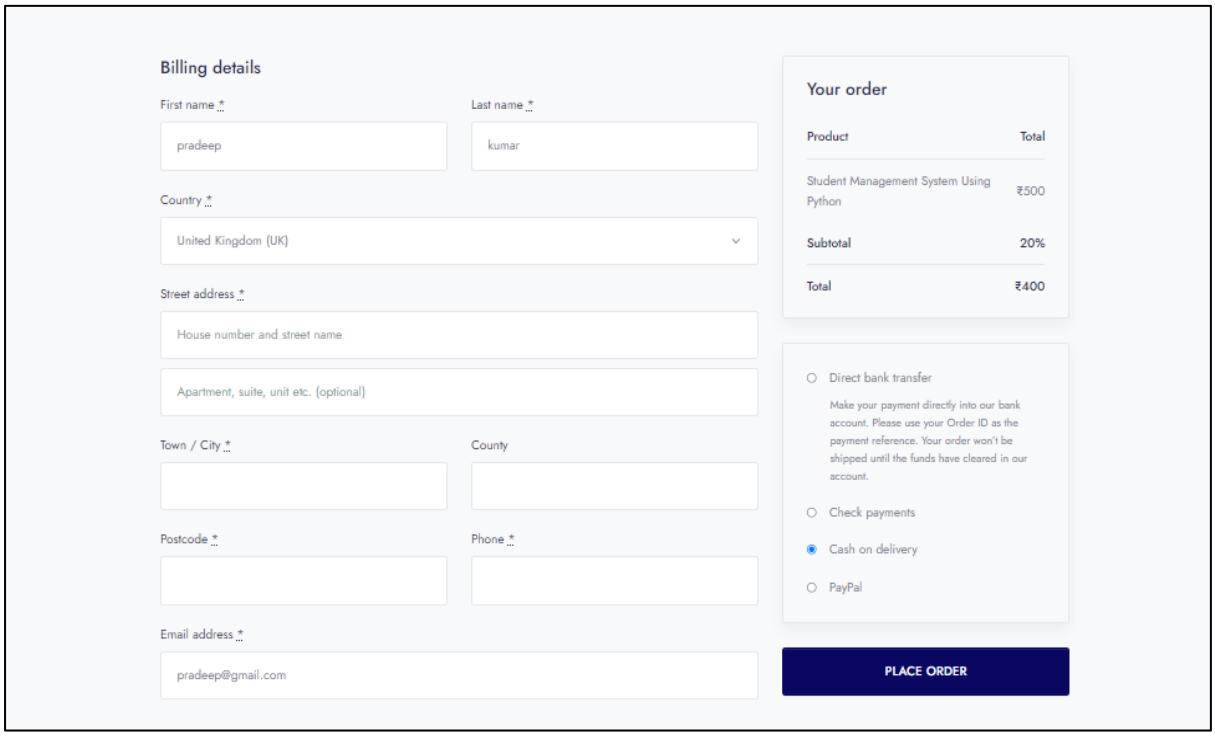
COURSE PAGE:



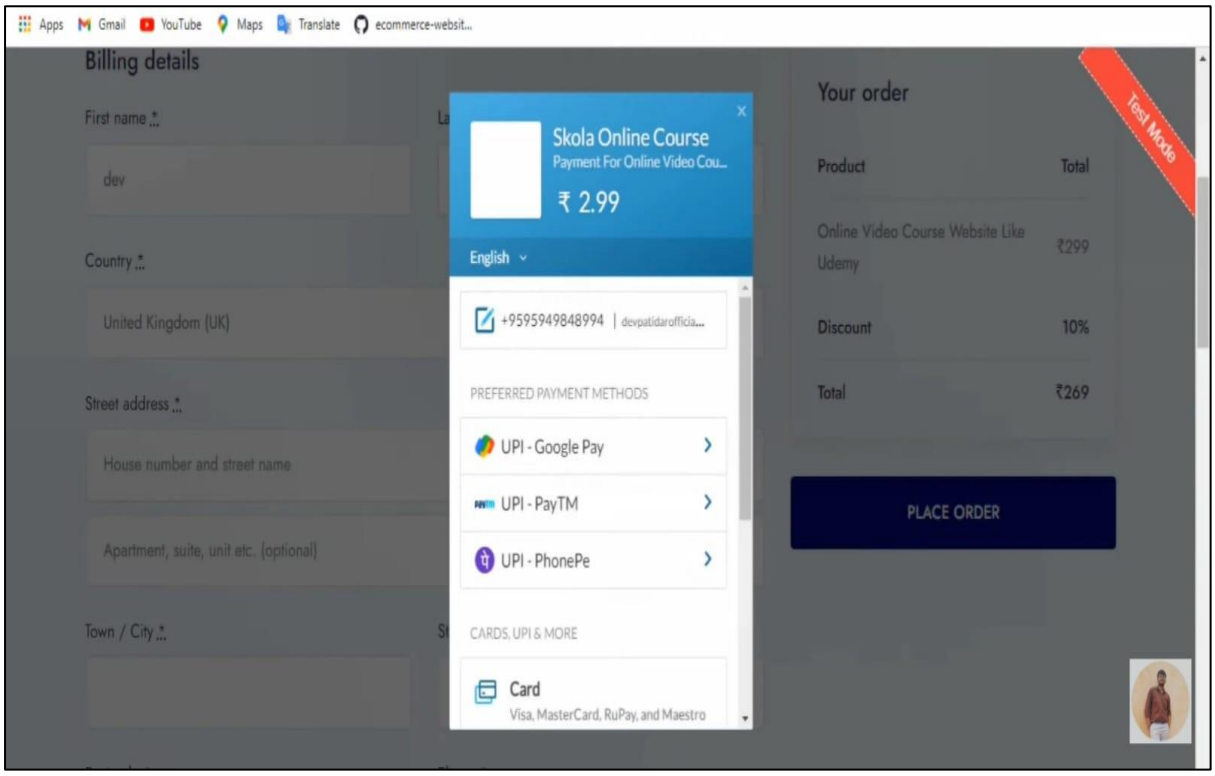
COURSE ENROLL PAGE:



COURSE PURCHASE PAGE:



PAYMENT PAGE:



B. SAMPLE CODING

Urls.py

```
from django.contrib import admin
from django.urls import path,include
from . import views,user_login
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('404',views.PAGE_NOT_FOUND,name= '404'),
    path('base/',views.BASE,name='base'),
    path('',views.HOME,name='home'),

    path('course',views.SINGLE_COURSE,name='single_course'),
    path('product/filter-data',views.filter_data,name="filter-data"),
    path('search',views.SEARCH,name='search'),
    path('course/<slug:slug>',views.COURSE_DETAILS,name='course_details'),
    path('contact',views.CONTACT_US,name='contact_us'),

    path('about',views.ABOUT_US,name='about_us'),
    path('accounts/register',user_login.REGISTER,name='register'),
    path('accounts/', include('django.contrib.auth.urls')),
    path('dologin',user_login.DO_LOGIN,name='dologin'),
   path('accounts/profile',user_login.PROFILE,name='profile'),
   path('accounts/profile/update',user_login.PROFILE_UPDATE,name='profile_update'),
   path('checkout/<slug:slug>',views.CHECKOUT,name='checkout')

]+ static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

Views.py

```
from django.shortcuts import redirect,render
from app.models import Categories,Course,Level,Video,UserCource
from django.template.loader import render_to_string
from django.http import JsonResponse
from django.db.models import Sum

def BASE(request):
    return render(request,'base.html')

def HOME(request):
    category=Categories.objects.all().order_by('id')[0:5]
    course = Course.objects.filter(status = 'PUBLISH').order_by('-id')
    context={'category':category,
            'course':course,}
    return render(request,'main/home.html',context)

def SINGLE_COURSE(request):
    category=Categories.get_all_category(Categories)
    level=Level.objects.all()
    course=Course.objects.all()
    FreeCoursecount=Course.objects.filter(price=0).count()
    PaidCoursecount=Course.objects.filter(price__gte=1).count()

    context={'category':category,'level':level,'course':course,
'FreeCoursecount':FreeCoursecount,
            ' PaidCoursecount': PaidCoursecount,}
    return render(request,'main/single_course.html',context)
```

```
def filter_data(request):
    category = request.GET.getlist('category[]')
    level = request.GET.getlist('level[]')
    price = request.GET.getlist('price[]')

    if price == ['pricefree']:
      course = Course.objects.filter(price=0)
    elif price == ['pricepaid']:
      course = Course.objects.filter(price__gte=1)
    elif price == ['priceall']:
      course = Course.objects.all()
    elif category:
      course = Course.objects.filter(category__id__in=category).order_by('-id')
    elif level:
      course = Course.objects.filter(level__id__in = level).order_by('-id')
    else:
      course = Course.objects.all().order_by('-id')
    context={
          'course':course,
          }
    t = render_to_string('ajax/course.html',context)
    return JsonResponse({'data': t})
def CONTACT_US(request):
   category=Categories.get_all_category(Categories)
   context={'category':category}
   return render(request,'main/contact.html',context)

def ABOUT_US(request):
   category=Categories.get_all_category(Categories)
   context={'category':category}
```

```
    return render(request,'main/about_us.html',context)
def SEARCH(request):
    query=request.GET['query']
    course=Course.objects.filter(title__icontains=query)
    context={'course':course}
    return render(request,'search/search.html',context)

def COURSE_DETAILS(request,slug):
    category=Categories.get_all_category(Categories)
time_duration=Video.objects.filter(course__slug=slug).aggregate(sum=Sum('time_duration')
    course_id=Course.objects.get(slug=slug)
    try:
        check_enroll=UserCource.objects.get(user=request.user,course= course_id)
    except UserCource.DoesNotExist:
        check_enroll= None

    course=Course.objects.filter(slug=slug)

    if course.exists():
        course=course.first()
    else:
        return redirect('404')

    context={'course':course,
            'category':category,
            'time_duration':time_duration,
            'check_enroll':check_enroll,
            }
    return render(request,'course/course_details.html',context)
```

```
def PAGE_NOT_FOUND(request):
  return render(request,'error/404.html')

def CHECKOUT(request,slug):
  course=Course.objects.filter(slug=slug)

  if course.price==0:
    course=UserCource(user=request.user,course=course)
    course.save()
    return redirect ('home')

  return render(request,"checkout/checkout.html")
```

Models.py -

```
from django.db import models
from django.contrib.auth.models import User
from django.utils.text import slugify
from django.db.models.signals import pre_save

# Create your models here.
class Categories(models.Model):
   icon = models.CharField(max_length=200,null=True)
   name = models.CharField(max_length=200)

   def __str__(self):
      return self.name

   def get_all_category(self):
      return Categories.objects.all().order_by('id')
```

```
class Author(models.Model):
    author_profile = models.ImageField(upload_to="Media/author")
    name = models.CharField(max_length=100, null=True)
    about_author = models.TextField()

    def __str__(self):
        return self.name

class Level(models.Model):
    name=models.CharField(max_length=100)

    def __str__(self):
        return self.name

class Language(models.Model):
    language=models.CharField(max_length=100)

    def __str__(self):
        return self.language


class Course(models.Model):
    STATUS = (
        ('PUBLISH','PUBLISH'),
        ('DRAFT', 'DRAFT'),
    )

    featured_image = models.ImageField(upload_to="Media/featured_img",null=True)
    featured_video = models.CharField(max_length=300,null=True)
```

```
    title = models.CharField(max_length=500)
    created_at = models.DateField(auto_now_add=True)
    author = models.ForeignKey(Author,on_delete=models.CASCADE,null=True)
    category = models.ForeignKey(Categories,on_delete=models.CASCADE)
    level=models.ForeignKey(Level,on_delete=models.CASCADE,null=True)
    description = models.TextField()
    price = models.IntegerField(null=True,default=0)
    discount = models.IntegerField(null=True)
    language= models.ForeignKey(Language,on_delete=models.CASCADE,null=True)
    deadline=models.CharField(max_length=100,null=True)
    slug = models.SlugField(default='', max_length=500, null=True, blank=True)
    status = models.CharField(choices=STATUS,max_length=100,null=True)
    Certificate=models.CharField(max_length=100,null=True)


    def __str__(self):
        return self.title
    def get_absolute_url(self):
        from django.urls import reverse
        return reverse("course_details", kwargs={'slug': self.slug})


def create_slug(instance, new_slug=None):
    slug = slugify(instance.title)
    if new_slug is not None:
        slug = new_slug
    qs = Course.objects.filter(slug=slug).order_by('-id')
    exists = qs.exists()
    if exists:
        new_slug = "%s-%s" % (slug, qs.first().id)
```

```
        return create_slug(instance, new_slug=new_slug)
    return slug
def pre_save_post_receiver(sender, instance, *args, **kwargs):
    if not instance.slug:
        instance.slug = create_slug(instance)
pre_save.connect(pre_save_post_receiver, Course)

class what_u_learn(models.Model):
    course=models.ForeignKey(Course,on_delete=models.CASCADE)
    points=models.CharField(max_length=1000)
    def __str__(self):
        return self.points

class Requirments(models.Model):
    course=models.ForeignKey(Course,on_delete=models.CASCADE)
    points=models.CharField(max_length=1000)
    def __str__(self):
        return self.points

class Lesson(models.Model):
    course=models.ForeignKey(Course,on_delete=models.CASCADE)
    name=models.CharField(max_length=200)
    def __str__(self):
        return self.name +"-"+self.course.title
class Video(models.Model):
    serial_number=models.IntegerField(null=True)
    thumbnail=models.ImageField(upload_to="Media/YT_Thumbnail",null=True)
    course=models.ForeignKey(Course,on_delete=models.CASCADE)
    lesson=models.ForeignKey(Lesson,on_delete=models.CASCADE)
    title=models.CharField(max_length=100)
```

```
    youtube_id=models.CharField(max_length=200)
    time_duration=models.IntegerField(null=True)
    preview=models.BooleanField(default=False)
    def __str__(self):
        return self.title

class UserCource(models.Model):
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    course=models.ForeignKey(Course,on_delete=models.CASCADE)
    paid=models.BooleanField(default=0)
    date=models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.name +"-"+self.course.title
```

Admin.py –

```
 from django.contrib import admin
from django.urls import path
from. import views
from.models import *

admin.site.register(Categories)
admin.site.register(Author)
admin.site.register(Course,course_admin)
admin.site.register(Level)
admin.site.register(what_u_learn)
admin.site.register(Requirments)
admin.site.register(Lesson)
admin.site.register(Video)
admin.site.register(Language)
admin.site.register(UserCource)
```