

**DIGITAL STUDY MATERIAL WEBSITE
PROJECT REPORT
SUBMITTED TO THE MADRAS UNIVERSITY IN PARTIAL
FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF BACHELOR OF COMPUTER SCIENCE
BY
M.ABINAYA (222201620)**

Under the guidance of
Mrs. V.G. PARASAKTHI, M.Sc, M.Phil.
DEPARTMENT OF COMPUTER SCIENCE



**DEPARTMENT OF COMPUTER SCIENCE
PROF.DHANAPALAN COLLEGE OF
SCIENCE & MANAGEMENT KELAMBAKKAM
APRIL - 2025**

PROF. DHANAPALAN COLLEGE OF SCIENCE & MANAGEMENT
KELAMBAKKAM
BSC-DEGREE
PROJECT REPORT
M.ABINAYA-222201620

BONAFIDE CERTIFICATE

This is to certify that this project work titled “**DIGITAL STUDY MATERIAL WEBSITE**” is Done by **M. ABINAYA** During the year 2024-2025.

This work is submitted for partial for the award of the degree of **BACHELOR OF COMPUTER SCIENCE** issued by university of Madras.

PROJECT GUIDE

HEAD OF THE DEPARTMENT

Submitted for the practical examination _____held on
_____Prof. Dhanapalan college of science and management,
Chennai.

Internal Examiner

External Examiner

DECLARATION

We hereby declare that the project **“DIGITAL STUDY MATERIAL”**.

Submitted by **M.ABINAYA** in partial fulfilment of the requirement for the degree of **BACHELOR OF COMPUTER SCIENCE** course of the university of madras is a record of the project work done by both originally.

(M.ABINAYA)

ACKNOWLEDGMENT

it is my grand pleasure to acknowledge the department of computer application prof. Dhanapalan College of Science and Management, kelambakkam for having assigned me the project work in DIGITAL STUDY MATERIAL.

I wish to extend my heart-felt gratitude respectful to the **CHAIRMAN, SECRETARY AND PRINCIPAL OF PROF. DHANAPALAN COLLEGE OF SCIENCE AND MANGEMENT, KELAMBAKKAM** for providing necessary facilities throughout of the courses

I express profound gratitude to my project guide Mrs. V.G. PARASAKTHI, M.Sc, M.Phil for her valuable guidance personal care encourage for completion of training Report Successfully.

At last but not least I would like to place on records my sincere thanks to all people who have enable me to completing my project report Successfully.

STATION:

Ms.M.ABINAYA

DATE:

ABSTRACT

- **The Digital Study Material system (developed using HTML, CSS, JavaScript, PHP, and SQL) is a web-based platform designed to provide structured and accessible study resources for computer science students. It allows users to read PDFs and take interactive quizzes, enhancing their learning experience.**
- **The Digital Study Material project helps students by offering well-organized content, reducing the need to search for scattered resources across multiple platforms. This ensures a seamless and efficient learning process.**
- **This system simplifies self-study and revision by integrating study materials and assessments in one place. Before this application, students had to rely on multiple sources, making learning time-consuming and inefficient. To overcome these challenges, we implement the Digital Study Material system.**

TABLE OF CONTENT

CHAPTER SNO	TITLE	PAGE NO
1.	INTRODUCTION 1.1 Problem statement 1.2 Solution statement 1.3 About solution tool 1.4 Future scope	1-4
2.	SYSTEM ANALYSIS	4-5
	2.1 Existing system	
	2.2 proposed system	
3.	REQUIREMENT SPECIFICATION	6-7
	3.1 Hardware specification	
	3.2 Software specification	
4.	SYSTEM DESIGN	8-14
	4.1 DFD Diagram	
	4.2 UML Diagram	
	4.4 ER Diagram	
5.	IMPLEMENTATION DETAIL	15-20
	5.1 Front end	
	5.2 Back end	
6.	TEST AND IMPLEMENTION	
7.	SCREEN INVOLVED IN APPLICATION	
	7.1 Brief explanation about the screens	33-34
8.	SOURCE CODE, SAMPLE SCREEN SHOT AND DATABASE 8.1 Source Code 8.2 Sample screen 8.3 Database	35- 55
9.	Conclusion	56

CHAPTER 1

INTRODUCTION

Introduction:

Digital study materials are electronic resources designed to support learning and academic achievement. They offer a range of benefits, including accessibility, convenience, and personalization. With digital study materials, students can access learning resources anywhere, anytime, and engage with interactive content, multimedia, and online communities. This can help to enhance the learning experience, improve outcomes, and increase flexibility and autonomy. Digital study materials can also provide real-time feedback, assessment, and tracking, enabling students to monitor their progress and identify areas for improvement. Additionally, digital study materials can facilitate collaboration, communication, and problem-solving skills, preparing students for success in an increasingly digital and interconnected world.

Problem Statement:

- The rise of digital education has made online study materials essential, but students often struggle to find well-organized and interactive learning resources.
- Many websites provide unstructured content, making it difficult for students to access relevant materials efficiently.
- The Digital Study Material Website aims to provide a centralized platform where students can read PDF-based study materials and take interactive quizzes to enhance their learning.

Solution Statement:

- The Digital Study Material Website is designed to streamline the process of accessing study materials and quizzes for computer science students.
- This platform provides structured PDFs for reading and interactive quizzes for self-assessment, making learning more effective.
- The system ensures an easy-to-use, responsive interface, developed using HTML, CSS, JavaScript, PHP, and SQL.

About Solution tool:

For the Digital Study Material Website, different tools are used for different components:

- Frontend: HTML, CSS, JavaScript – for user interface and interactive elements.
- Backend: PHP – for processing user requests and quiz handling.
- Database: SQL – for storing PDFs, quizzes, and user progress.

Future Scope:

- The Digital Study Material Website has a promising future scope with potential enhancements and expansion to further improve efficiency.
- The platform aims to provide users with downloadable PDF resources, conduct interactive quizzes to test knowledge, and continuously add new questions to ensure an engaging and comprehensive learning experience.

CHAPTER 2

SYSTEM ANALYSIS

SYSTEM ANALYSIS:

System analysis involves examining and understanding the existing system, identifying its limitations, and determining the requirements for a new system.

PROPOSED ANALYSIS:

The proposed system aims to provide a more efficient, effective, and user-friendly solution, leveraging modern technologies and design principles to meet the evolving needs of users.

EXISTING SYSTEM:

The existing system is a traditional, manual-based approach that relies heavily on paper-based documents and face-to-face interactions, resulting in inefficiencies, errors, and limited accessibility.

CHAPTER 3

REQUIREMENT SPECIFICATION

HARDWARE SPECIFICATION:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-Section discuss the various aspects of hardware requirements.

Hardware requirement for present project:

- Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
1.80 GHz
- Ram: 8.00 GB
- Hard disk: 500 GB

SOFTWARE SPECIFICATION:

- software Requirements deal with defining software resource requirements and per- requisite that need to be installed on a computer to provide optimal functioning of an application.
- These requirements or per-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

software requirement for present project:

- Operating system: windows 10 pro
- Front end: HTML, CSS, JAVASCRIPT
- Back end: PHP
- Data base: MySQL

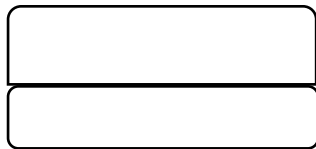
CHAPTER 4

SYSTEM DESIGN

DATA FLOW DIAGRAM:

A data flow diagram is graphical tool used to describe and analyses of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independent of physical components associated with the system. These are known as the logical data flow diagrams.

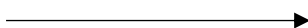
DFD SYMBOLS:



Process that transforms data flow



Source of destination of data



Data Flow



Data Store

USE CASE DIAGRAM:

- A Digital Study Material Website's can use case diagram

Applications

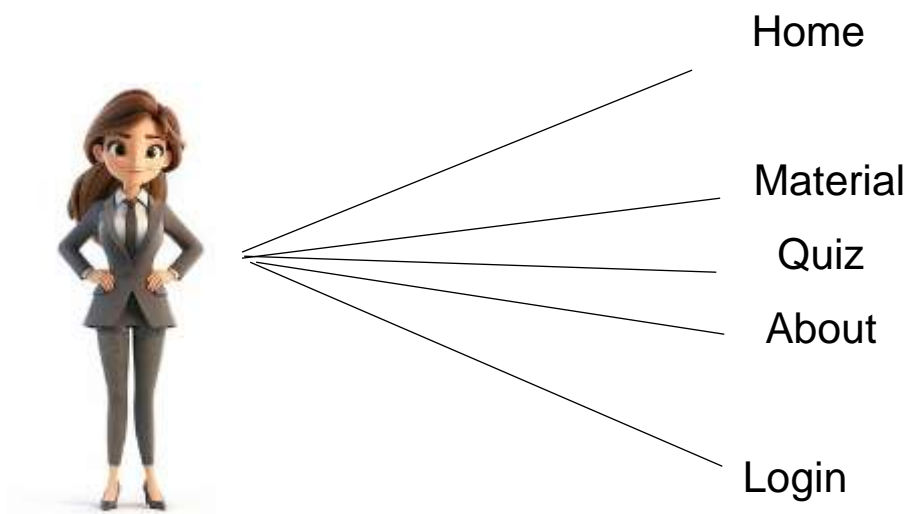
Digital study materials offer various applications that enhance learning and accessibility. Some key applications include:

1. E-Learning Platforms – Used in online courses, MOOCs (Massive Open Online Courses), and virtual classrooms for remote education.
2. Self-Paced Learning – Allows students to study at their convenience with PDFs, videos, and interactive quizzes.
3. Exam Preparation – Digital notes, flashcards, and mock tests help students prepare efficiently for competitive exams.
4. Collaborative Learning – Enables sharing of study materials and discussions through cloud storage and online forums.
5. Cost-Effective Education – Reduces the need for physical textbooks, making education more affordable.
6. Accessibility & Inclusivity – Beneficial for students with disabilities through text-to-speech, subtitles, and screen readers.
7. Real-Time Updates – Digital materials can be updated instantly, ensuring students access the latest content.
8. Performance Tracking – Helps educators monitor student progress through analytics and adaptive learning.

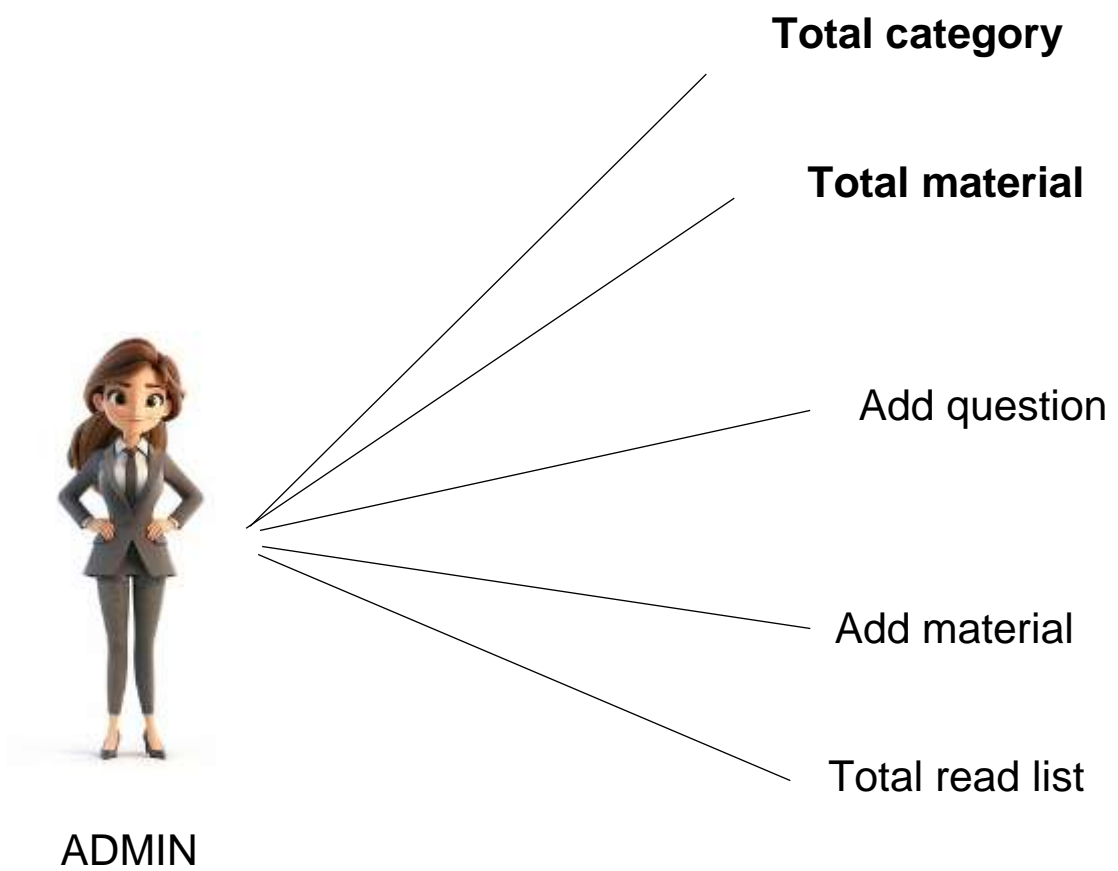
UML DIAGRAM:

It represents the flow of the data in the application and by analyses this, one is able to understand the system completely, by graphical representation such as circle, rectangle, square etc. It allows us to get what you want.

User:



Admin:



ER DIAGRAM:

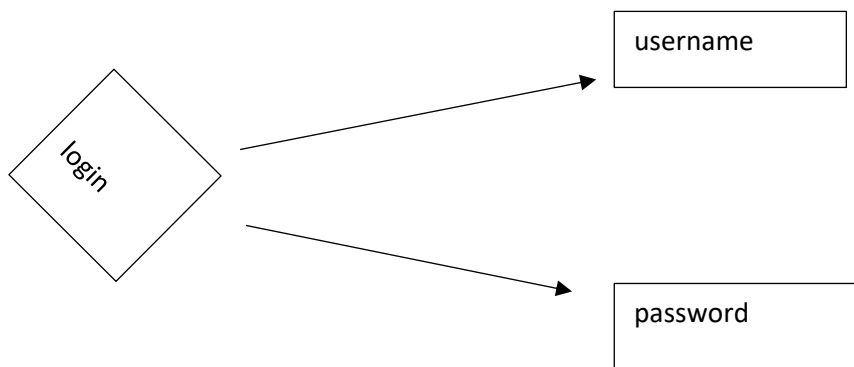
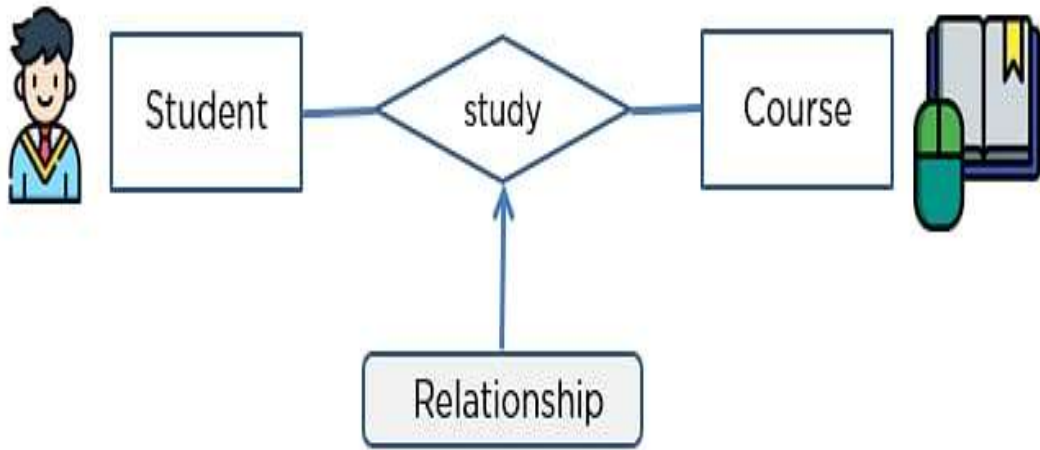
- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnections of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

Uses of entity relationship diagram:

1) Database design: ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

2) Database troubleshooting: ER diagrams are used to analyse existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.

3) Business information systems: The diagrams are used to design or analyzing relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.



CHAPTER 5

IMPLEMENTATION DETAILS

IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

5.1 FRONT END

5.1.1 HTML



- Hypertext Markup Language (HTML) 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. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
- 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 text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

5.1.2 CSS



- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content.
- CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used, for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.
- One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site.

5.1.3 JAVASCRIPT:



JavaScript is a high-level, dynamic, and interpreted programming language that is primarily used for client-side scripting on the web. It allows developers to create interactive web pages, web, and mobile applications.

Some key features of JavaScript include:

1. **Dynamic typing:** JavaScript is dynamically typed, which means that variable types are determined at runtime.
2. **First-class functions:** JavaScript functions are first-class citizens, which means they can be passed as arguments to other functions, returned from functions, and stored in data structures.
3. **Prototype-based applications inheritance:** JavaScript uses a prototype-based inheritance model, which allows objects to inherit properties and behavior from other objects.
4. **Async programming:** JavaScript provides built-in support for asynchronous programming through callbacks, promises, and async/await.

BACK END

5.2.1. PHP:



- PHP (Hypertext Processor) is a general-purpose scripting language and interpreter that is freely available and widely used for web development. The language is used primarily for server-side scripting, although it can also be used for command-line scripting and, to a limited degree, desktop applications.
- It is primarily used for server-side scripting, but can also be used for command-line scripting and desktop applications.

The biggest strength of PHP:

Scalability: PHP is a good choice for large-scale projects.

Community support: PHP has a strong community of support.

Ease of use: PHP has a simple syntax that is user-friendly and ideal for beginners.

Integration: PHP integrates with various databases, including MySQL.

Stability: PHP is stable for a few years with regular support for various versions.

Reuse: PHP helps with reusing code, so you don't have to write lengthy code.

Manage code: PHP helps with managing code.

5.2.2. MySQL:



MySQL, or My Structured Query Language, is a free, open-source relational database management system (RDBMS) that stores data in tables instead of one big storage room. It uses Structured Query Language (SQL) and is developed by Oracle.

► MySQL is used for storing and managing data. For example, a database can be a shopping list, picture gallery, or corporate network. The combination of PHP and MySQL, can be used to create a variety of websites, from contact forms to corporate portals.

Keywords are words that have significance in SQL. Certain keywords, such as SELECT, DELETE, or BIGINT, are reserved and require special treatment for use as identifiers such as table and column names. This may also be true for the names of built-in functions.

CHAPTER 6

TESTING AND IMPLEMENTATION

6. TESTING AND IMPLEMENTATION

- The term implementation has different meanings ranging from the conversion of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same, Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement

6.1 UNIT TESTING

6.1.1 Introduction

- In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

6.1.2 Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

1) Find problems early: Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

2) Facilitates Change: Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g.. in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

3) Simplifies Integration: Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

4) Documentation: Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API). Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviour that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

6.2 INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

6.2.1 Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activation, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the chromatically complexity of the software and software architecture, re usability of modules and life-cycle and versioning management. Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

6.3 SOFTWARE VERIFICATION AND VALIDATION

6.3.1 Introduction

- In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfil its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development life cycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between:
 - Validation: Are we building the right product?
 - Verification: Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1):

- **Software Verification:** The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
- **Software Validation:** The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing".
- **Malfunction** - according to its specification the system does not meet its specified functionality. Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required. Within the modeling and simulation (M&S) community,

the definitions of verification, validation and accreditation are similar.

- M&S Verification is the process of determining that a computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
- M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

6.3.2 Classification of Methods:

In mission-critical software systems, where flawless performance is absolutely necessary. formal methods may be used to ensure the correct operation of a system.

6.3.3 Test Cases

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

6.4 Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well.

6.4.1 Test Procedures

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

6.4.2 Test Cases:

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used.

6.5 White-Box Testing

White box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal strictures or workings of an application, as opposed to its functionality (Le. black box testing) in white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, eg. in-circuit testing (ICT) White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. I can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

6.5.1 Levels

1. Unit testing: White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects clearly on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

2. **Integration testing:** White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
3. **Regression testing:** White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

6.5.2 Procedures

- White-box testings basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:
- Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information. Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly. Output involves preparing final report that encompasses all of the above preparations and results.

6.5.3 Advantages

- White-box testing is one of the two biggest testing methodologies used today. It has several major advantages: Side effects of having the knowledge of the source code is beneficial to thorough testing. Optimization of code by revealing hidden errors and being able to remove these possible defects. Gives the programmer introspection because developers carefully describe any new implementation. Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.

6.5.4 Disadvantages

- Although white-box testing has great advantages, it is not perfect and contains some disadvantages: White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done. On some occasions, it is not realistic to be able to test every single existing conditions will be untested. The tests focus on the software as it exists, and missing functionality may not be discovered.

6.6 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

CHAPTER 7
SCREEN INVOLVED IN APPLICATION

7. Screen involved in Application:

User

- Login
- My Account
- Download Material
- Attend Quiz
- Add Quiz

Admin

- Admin Login
- Insert and View all Material
- Insert and View all Categories
- Insert and View all Quiz
- View Student list
- Create new Feature

7.1 Brief explanation about the screens:

7.1.1 Login:

It allows us to get into the system by gathering username and password.

7.1.2 Register:

A register portal is a page that asks for a user's username, email, password and then sends the data to the database.

7.1.3 User home page:

The user's homepage contain the all packages, my account, shopping cart, contact us, register and the user can also view as a guest.

7.1.4 Contact Us:

The user's can contact to the developers via contact us page. the user can be visit full detail in contact us page.

7.1.5 Admin Dashboard:

Administrators to manage and monitor various aspects of a websites, application or business. It viewing important information, such as website user accounts and data analytic.

7.1.6 Logout:

By using these features, user are able to logout from the system whenever they wish to leave the system.

7.1.7 Material:

The Material Module provides a comprehensive collection of study resources, where users can download PDFs and read them at their convenience to supplement their learning.

Material category Adding Programming Language, Framework, Database, Interview Question, Adavnced technology.

7.1.8 Quiz:

The Quiz Module enables administrators to conduct quizzes and add new questions, while allowing users to participate in quizzes and receive instant feedback on their performance.

7.1.9 Add Question:

To add more questions, simply select the "Add Question" option, fill in the required details, and submit - your new question will be instantly added to the quiz database.

CHAPTER 8

Source Code, Sample Screenshot & Database

8.1 Source Code:

8.1.1 INDEX:

```
<?php

    include("connection.php");

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Email and Password Verification</title>

</head>
<body>
    <div class="form-container">
        <h2>Login</h2>
        <form id="loginForm">
            <div class="form-group">
                <label for="email">Email:</label>
                <input type="email" id="email" placeholder="Enter your email"
required>
                <span id="emailError" class="error"></span>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" placeholder="Enter your password"
required>
                <span id="passwordError" class="error"></span>
            </div>
            <button type="submit">Login</button>
        </form>
        <div id="successMessage" class="success"></div>
    </div>
    <style>
        /* General styles */
body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to bottom, pink, white);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
```

```

.form-container {
  background: #fff;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  width: 300px;
}

h2 {
  text-align: center;
  margin-bottom: 20px;
}

.form-group {
  margin-bottom: 15px;
}

label {
  display: block;
  margin-bottom: 5px;
}

input {
  width: 100%;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 4px;
  font-size: 14px;
}

button {
  width: 100%;
  padding: 10px;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

.error {
  color: red;
  font-size: 12px;
  margin-top: 5px;
}

```

```

}

        .success {
        color: green;
        text-align: center;
        margin-top: 10px;
    }
</style>

<script>
document.getElementById("loginForm").addEventListener("submit", function (e) {
    e.preventDefault();

    // Clear previous messages
    document.getElementById("emailError").textContent = "";
    document.getElementById("passwordError").textContent = "";
    document.getElementById("successMessage").textContent = "";

    // Get input values
    const email = document.getElementById("email").value.trim();
    const password = document.getElementById("password").value.trim();

    let isValid = true;

    // Email validation
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(email)) {
        document.getElementById("emailError").textContent = "Invalid email
format.";
        isValid = false;
    }

    // Password validation (at least 8 characters, 1 number)
    const passwordRegex = /^(?=.*[0-9]).{8,}$/;
    if (!passwordRegex.test(password)) {
        document.getElementById("passwordError").textContent =
        "Password must be at least 8 characters long and contain at least 1
number.";
        isValid = false;
    }

    // If valid, show success and redirect
    if (isValid) {
        document.getElementById("successMessage").textContent = "Login
successful!";
        setTimeout(() => {
            window.location.href = "welcome.php"; // Redirect to the homepage
        }, 1000);
    }
}

```

```

    }
  });
</script>

</body>
</html>

```

8.1.2 HOMEPAGE

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Digital Study Materials</title>
  <link rel="stylesheet" href="work.css">
</head>
<body>
  <header>
    <div class="logo">Digital Study Materials</div>
    <nav>
      <ul class="nav-links">
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#materials">Materials</a></li>
        <li><a href="#quiz">Quizzes</a></li>
        <li><a href="#contact">Contact</a></li>
        <li><a href="index.php">Login</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="home" class="hero-section">
      <h1>Welcome to Digital Study Materials</h1>
      <p>Explore a world of knowledge at your fingertips. Access study
materials and test your skills with interactive quizzes!</p>
      <button id="get-started">Get Started</button>
    </section>

    <section id="about">
      <h2>About This Project</h2>

```

```

    <p>This platform provides digital study materials specifically designed
    for Computer Science students. Enhance your learning experience with curated
    PDFs and engaging quizzes.</p>
    </section>

    <section id="materials">
        <h2>Study Materials</h2>
        <p>Access high-quality, topic-specific PDFs to master key concepts. From
        programming languages to algorithms, we've got you covered!</p>
        <a href="material.php"><button>View Materials</button></a>
    </section>

    <section id="quiz">
        <h2>Take a Quiz</h2>
        <p>Challenge yourself with quizzes based on the study materials. Test
        your understanding and track your progress!</p>
        <a href="quizes.php"><button>Start Quiz</button></a>
    </section>

    <section id="contact">
        <h2>Contact Us</h2>
        <p>Have questions or feedback? Reach out to us!</p>
        <form>
            <input type="text" placeholder="Your Name" required>
            <input type="email" placeholder="Your Email" required>
            <textarea placeholder="Your Message" required></textarea>
            <button type="submit">Submit</button>
        </form>
    </section>
</main>

<footer>
    <p>&copy; 2025 Digital Study Materials. All rights reserved.</p>
</footer>

<style>
/* Reset and global styles */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: #333;
}

```

```

header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px 20px;
  background-color: #4caf50;
  color: #fff;
}

header .logo {
  font-size: 24px;
  font-weight: bold;
}

.nav-links {
  list-style: none;
  display: flex;
  gap: 15px;
}

.nav-links a {
  color: #fff;
  text-decoration: none;
}

.hero-section {
  text-align: center;
  padding: 50px 20px;
  background: linear-gradient(to right, #4facfe, #00f2fe);
  color: #fff;
}

.hero-section h1 {
  font-size: 2.5rem;
  margin-bottom: 15px;
}

.hero-section p {
  font-size: 1.2rem;
  margin-bottom: 20px;
}

button {
  padding: 10px 20px;
  background-color: #fff;
  border: none;
  cursor: pointer;
}

```

```

    font-size: 1rem;
    color: #333;
}

button:hover {
    background-color: #ddd;
}

section {
    padding: 20px;
    text-align: center;
}

footer {
    text-align: center;
    padding: 10px;
    background-color: #4caf50;
    color: #fff;
}
</style>
<script>
    function toggleSubModules(id) {
        const element = document.getElementById(id);
        const button = element.previousElementSibling.querySelector('.toggle-btn');

        if (element.classList.contains('hidden')) {
            element.classList.remove('hidden');
            button.textContent = '-';
        } else {
            element.classList.add('hidden');
            button.textContent = '+';
        }
    }
</script>
</body>
</html>

```

8.1.3 MATERIAL

```

<!DOCTYPE html>
<html lang="en">
<head>

```



```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Digital Study Materials</title>
<link rel="stylesheet" href="work.css">
</head>
<body>
  <header>
    <div class="logo">Digital Study Materials</div>
    <nav>
      <ul class="nav-links">
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#materials">Materials</a></li>
        <li><a href="#quiz">Quizzes</a></li>
        <li><a href="#contact">Contact</a></li>
        <li><a href="index.php">Login</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="home" class="hero-section">
      <h1>Welcome to Digital Study Materials</h1>
      <p>Explore a world of knowledge at your fingertips. Access study materials and test your skills with interactive quizzes!</p>
      <button id="get-started">Get Started</button>
    </section>

    <section id="about">
      <h2>About This Project</h2>
      <p>This platform provides digital study materials specifically designed for Computer Science students. Enhance your learning experience with curated PDFs and engaging quizzes.</p>
    </section>

    <section id="materials">
      <h2>Study Materials</h2>
      <p>Access high-quality, topic-specific PDFs to master key concepts. From programming languages to algorithms, we've got you covered!</p>
      <a href="material.php"><button>View Materials</button></a>
    </section>

    <section id="quiz">
      <h2>Take a Quiz</h2>
      <p>Challenge yourself with quizzes based on the study materials. Test your understanding and track your progress!</p>
      <a href="quizes.php"><button>Start Quiz</button></a>
    </section>
  </main>

```

```

<section id="contact">
  <h2>Contact Us</h2>
  <p>Have questions or feedback? Reach out to us!</p>
  <form>
    <input type="text" placeholder="Your Name" required>
    <input type="email" placeholder="Your Email" required>
    <textarea placeholder="Your Message" required></textarea>
    <button type="submit">Submit</button>
  </form>
</section>
</main>

<footer>
  <p>&copy; 2025 Digital Study Materials. All rights reserved.</p>
</footer>

<style>
/* Reset and global styles */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333;
}

header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px 20px;
  background-color: #4caf50;
  color: #fff;
}

header .logo {
  font-size: 24px;
  font-weight: bold;
}

.nav-links {
  list-style: none;
  display: flex;

```

```

    gap: 15px;
}

.nav-links a {
  color: #fff;
  text-decoration: none;
}

.hero-section {
  text-align: center;
  padding: 50px 20px;
  background: linear-gradient(to right, #4facfe, #00f2fe);
  color: #fff;
}

.hero-section h1 {
  font-size: 2.5rem;
  margin-bottom: 15px;
}

.hero-section p {
  font-size: 1.2rem;
  margin-bottom: 20px;
}

button {
  padding: 10px 20px;
  background-color: #fff;
  border: none;
  cursor: pointer;
  font-size: 1rem;
  color: #333;
}

button:hover {
  background-color: #ddd;
}

section {
  padding: 20px;
  text-align: center;
}

footer {
  text-align: center;
  padding: 10px;
  background-color: #4caf50;
  color: #fff;
}

```

```

}
</style>
<script>
    function toggleSubModules(id) {
        const element = document.getElementById(id);
        const button = element.previousElementSibling.querySelector('.toggle-btn');

        if (element.classList.contains('hidden')) {
            element.classList.remove('hidden');
            button.textContent = '-';
        } else {
            element.classList.add('hidden');
            button.textContent = '+';
        }
    }
</script>
</body>
</html>

```

8.1.4 CONNECTION

```

<?php
    $servername = "localhost";
    $username = "root";
    $password = "";
    $db_name = "database1";
    $conn = new mysqli($servername, $username, $password, $db_name, 3306);
    if($conn->connect_error){
        die("Connection failed". $conn->connect_error);
    }
    echo "";

?>

```

8.1.5 QUIZ

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quiz with Add Question Feature</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;

```

```

        background-color: #f4f4f4;
    }
    .quiz-container, .add-question-container {
        max-width: 600px;
        margin: 20px auto;
        background: #fff;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    }
    button {
        width: 100%;
        padding: 10px;
        margin: 10px 0;
        background-color: #007BFF;
        color: white;
        border: none;
        border-radius: 5px;
        font-size: 16px;
        cursor: pointer;
    }
    button:hover {
        background-color: #0056b3;
    }
    .result {
        text-align: center;
        margin-top: 20px;
        font-size: 18px;
        font-weight: bold;
    }
</style>
</head>
<body>
    <div id="quiz-container" class="quiz-container">
        <h1>Loading Quiz...</h1>
    </div>

    <div class="add-question-container">
        <h2>Add a New Question</h2>
        <form id="add-question-form">
            <label for="category">Category:</label>
            <select id="category" required>
                <option value="html">HTML</option>
                <option value="css">CSS</option>
                <option value="javascript">JavaScript</option>
                <option value="Python">Python</option>
                <option value="C">C</option>
                <option value="C++">C++</option>
            </select>
        </form>
    </div>

```

```

        <option value="Artificial intelligence">Artificial
Intelligence </option>
        <option value="Mobile Computing">Mobile Computing </option>
        <option value="Cloud Computing">Cloud Computing </option>
        <option value="Computer Network">Computer Network</option>
        <option value="MySQL">MySQL</option>
        <option value="Flask">Flask</option>
        <option value="Django">Django</option>
        <option value="vb.net">vb.net</option>
    </select>
    <br><br>
    <label for="question">Question:</label>
    <input type="text" id="question" required><br><br>

    <label for="option1">Option 1:</label>
    <input type="text" id="option1" required><br><br>

    <label for="option2">Option 2:</label>
    <input type="text" id="option2" required><br><br>

    <label for="option3">Option 3:</label>
    <input type="text" id="option3" required><br><br>

    <label for="option4">Option 4:</label>
    <input type="text" id="option4" required><br><br>

    <label for="correct">Correct Option (1-4):</label>
    <input type="number" id="correct" min="1" max="4"
required><br><br>

    <button type="button" onclick="addQuestion()">Add
Question</button>
    </form>
</div>

<script>
    const quizzes = {
        html: [
            { question: "What does HTML stand for?", options: ["Hyper Text
Markup Language", "High Text Machine Language", "Hyper Tool Markup Language"],
correct: 0 },
            { question: "Which tag is used to define a hyperlink?",
options: ["link", "a", "href"], correct: 1 },
            {question: "css language which tag to connect a html?",
options: ["a","link","h1","br"], correct: 2 },
            {question: "which of the Following is used to define a
navigation block in HTML5?",
options:["header","nav","section","footer"],correct:2},

```

```

        {question: "which HTML element is used to Display image on a
web page?",options:["img","picture","image","media"],correct:1},
        {question: "what is the correct HTML tag for inserting a line
break?",options:["break","br","newline","lb"],correct:2},
    ],
    css: [
        { question: "What does CSS stand for?", options: ["Cascading
Style Sheets", "Computer Style Sheets", "Creative Style Sheets"], correct: 0
    },
    ],
    javascript: [
        { question: "Which of the following is used to declare a
variable in JavaScript?", options: ["var", "let", "const", "All of the
above"], correct: 3 },
    ],
    python: [
        { question:"who introduced python?", options:["marking
jackson","guido van rosum","Bjarne stroustrup","brendan Eich"],correct:0},
        {question: "What is the difference between static and dynamic
typing in Python?",
options: ["Static typing is used for variables, dynamic typing is used for
functions", "Static typing is used for functions, dynamic typing is used for
variables", "Python uses static typing", "Python uses dynamic typing"],
correct: 3
    },
    {
question: "How do you handle errors in Python?",
options: ["Using try-except blocks", "Using if-else statements", "Using for
loops", "Using while loops"],
correct: 0
    },
    {
question: "What is the purpose of the 'self' parameter in Python classes?",
options: ["To refer to the instance of the class", "To refer to the class
itself", "To refer to the parent class", "To refer to the child class"],
correct: 0
    },
    ]

    };

const urlParams = new URLSearchParams(window.location.search);
const category = urlParams.get("category");

function renderQuiz(category) {
    if (category && quizzes[category]) {
        const quiz = quizzes[category];
        let html = `

## 


```

```

        quiz.forEach((q, i) => {
            html += `<div>
                <h3>${i + 1}. ${q.question}</h3>
                ${q.options
                    .map((opt, index) => `<label><input type="radio"
name="q${i}" value="${index}"> ${opt}</label><br>`)
                    .join("")}
            </div>`;
        });
        html += `<button onclick="submitQuiz('${category}')">Submit
Quiz</button>`;
        document.getElementById("quiz-container").innerHTML = html;
    } else {
        document.getElementById("quiz-container").innerHTML =
`<h2>Invalid Category</h2>`;
    }
}

function submitQuiz(category) {
    const quiz = quizzes[category];
    let score = 0;
    quiz.forEach((q, i) => {
        const selected =
document.querySelector(`input[name="q${i}"]:checked`);
        if (selected && parseInt(selected.value) === q.correct) {
            score++;
        }
    });
    const result = `You scored ${score} out of ${quiz.length}!`;
    document.getElementById("quiz-container").innerHTML += `<div
class="result">${result}</div>`;
}

function addQuestion() {
    const category = document.getElementById("category").value;
    const question = document.getElementById("question").value;
    const options = [
        document.getElementById("option1").value,
        document.getElementById("option2").value,
        document.getElementById("option3").value,
        document.getElementById("option4").value,
    ];
    const correct = parseInt(document.getElementById("correct").value)
- 1;

    if (category && question && options.length === 4 &&
!isNaN(correct)) {
        quizzes[category].push({ question, options, correct });
    }
}

```

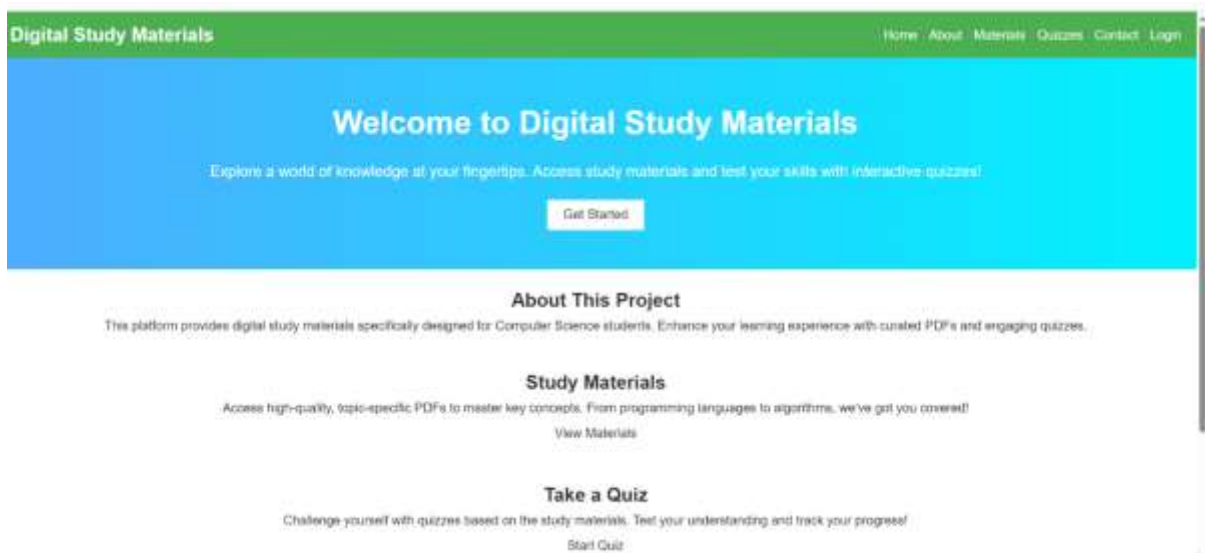
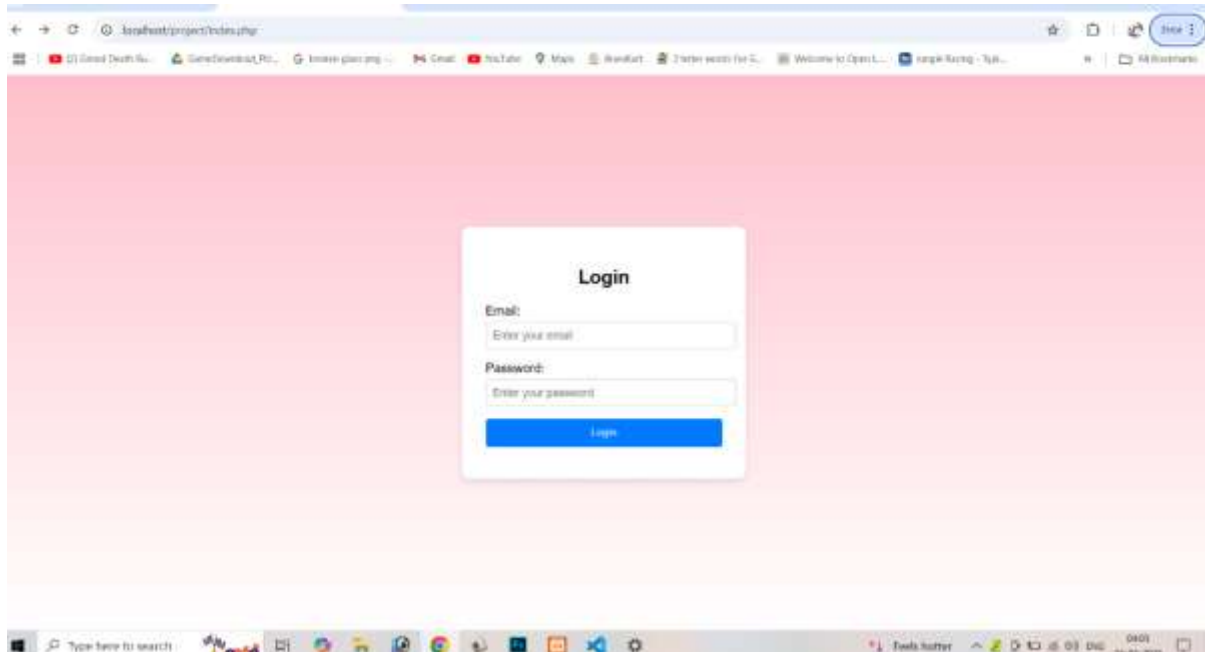


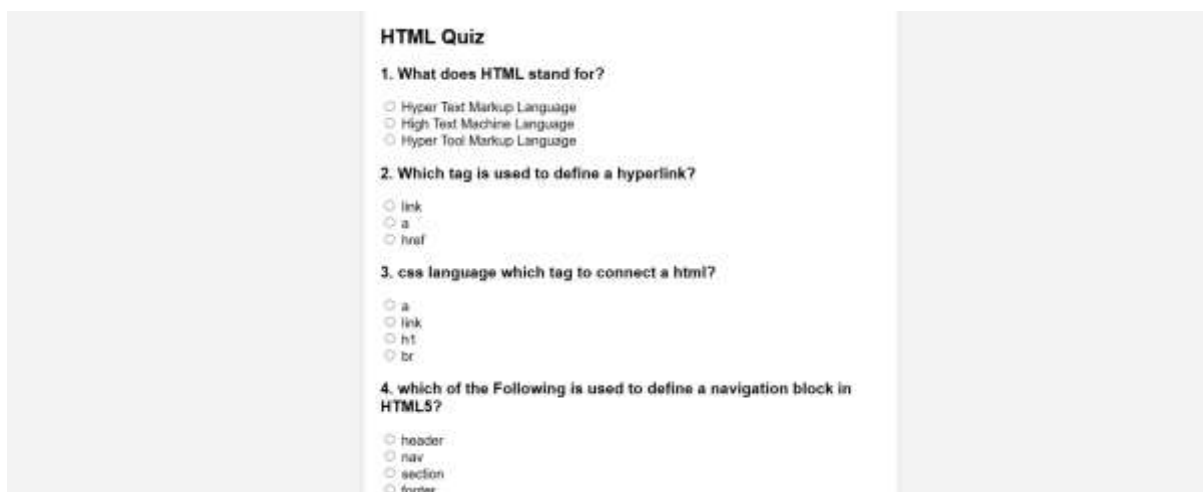
```
        alert("Question added successfully!");
        document.getElementById("add-question-form").reset();
        renderQuiz(category);
    } else {
        alert("Please fill out all fields correctly.");
    }
}

    if (category) renderQuiz(category);
</script>
</body>
</html>
```

CHAPTER 7

OUTPUT





Add a New Question

Category:

Question:

Option 1:

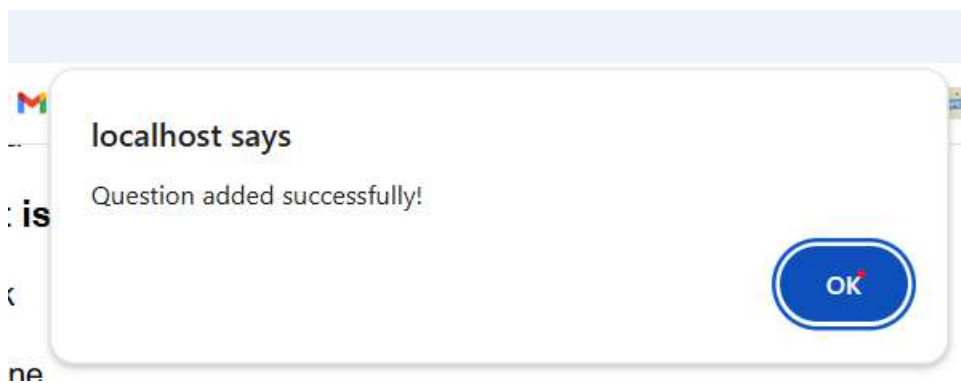
Option 2:

Option 3:

Option 4:

Correct Option (1-4):

Add Question



CONCLUSION:

The digital study material provides an innovative and interactive learning experience, offering numerous benefits, including:

- **Accessibility:** Available anywhere, anytime.
- **Personalization:** Catering to individual learning styles.
- **Engagement:** Interactive features, such as quizzes and multimedia.
- **Cost-effectiveness:** Reducing textbook and travel costs.

By leveraging digital study materials, learners can:

- Improve understanding and retention.
- Develop critical thinking and problem-solving skills.
- Enhance collaboration and communication.

Embracing digital study materials paves the way for a more efficient, effective, and enjoyable learning experience.