**A PROJECT REPORT ON**

**ISCHOOL (E-LEARNING MANAGEMENT SYSTEM)**

**DEVELOPED FOR**

**UNIFIED MENTOR, GURUGRAM**

**BY**

**PRIYANKA DILIP MARATHE**

**FOR THE PARTIAL FULFILMENT OF THE DEGREE**

**INTEGRATED MASTER OF COMPUTER APPLICATION**

**SUBMITTED TO**

****

**KCES’S INSTITUTE MANAGEMENT AND RESEARCH,**

**JALGAON**

**AFFILIATED TO**

**KBC NORTH MAHARASHTRA UNIVERSITY, JALGAON**

**2024 – 2025**

**A PROJECT REPORT ON**

**ISCHOOL (E-LEARNING MANAGEMENT SYSTEM)**

**DEVELOPED FOR**

**UNIFIED MENTOR, GURUGRAM**

**BY**

**PRIYANKA DILIP MARATHE**

**FOR THE PARTIAL FULFILMENT OF THE DEGREE**

**INTEGRATED MASTER OF COMPUTER APPLICATION**

**SUBMITTED TO**

****

**KCES’S INSTITUTE MANAGEMENT AND RESEARCH,**

**JALGAON**

**AFFILIATED TO**

**KBC NORTH MAHARASHTRA UNIVERSITY, JALGAON**

**2024 – 2025**

CERTIFICATE

To whomsoever it may concern

This is to certify that Ms. **Priyanka Dilip Marathe** a student of Integrated MCA (Master of Computer Application) from Institute of Management and Research, Jalgaon has completed the full time Industrial Training with project titled **ISchool (E-Learning Management System)** at **Unified Mentor Pvt. Ltd**, **Gurugram.**

She has submitted satisfactory report in partial fulfilment of the requirement for the award of the degree of Integrated Master of Computer Application (MCA) during academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Ms. Deepali Kirange |  | Ms. Deepali Kirange |
| Internal Guide |  | Head, Dept. of IMCA |
|  |  |  |
| Dr. Varsha Pathak |  | Prof. B. V. Pawar |
| Dean, School of Computer Applications |  | Director, IMR Jalgaon |
|  |  |  |
| External Examiner |  | External Examiner |

**INTERNSHIP CERTIFICATE**



**ACKNOWLEDGEMENT**

I express a deep sense of gratitude towards Drishti Madaan HR mam for providing me with the opportunity to work with “Unified Mentor Pvt Ltd, Gurugram”.

I express my sincere thanks to Ms. Deepali Kirange Internal Project Guide for the encouragement, insistence, assistance, and valuable guidance for completion of this project.

I would like to give special thanks to Director Prof. B. V. Pawar Sir, Head, Dept. of IMCA Ms. Deepali Kirange Mam and all my faculty members who has been constant source of inspiration to me throughout my completion of IMCA at KCES’s Institute of Management and Research, Jalgaon.

Last but not the least I would like to thanks my family and friends for being so supportive and generous to me throughout my learning days.

* *Priyanka Dilip Marathe*

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Topic** | | **Page No.** |
| **1** | **INTRODUCTION** | |  |
|  | **1.1** | Company Profile |  |
|  | **1.2** | Introduction to Project |  |
| **2** | **SYSTEM REQUIREMENTS ANALYSIS** | |  |
|  | **2.1** | Identification of need |  |
|  | **2.2** | Software Requirement Specification |  |
|  | **2.3** | Data Gathering |  |
|  | **2.4** | Software Process Model |  |
| **3** | **FEASIBILITY STUDY** | |  |
|  | **3.1** | Economic Feasibility |  |
|  | **3.2** | Technical Feasibility |  |
|  | **3.3** | Behavioural Feasibility |  |
|  | **3.4** | Time Feasibility |  |
|  | **3.5** | Operational Feasibility |  |
| **4** | **SYSTEM DESIGN** | |  |
|  | **4.1** | Use Case Diagram |  |
|  | **4.2** | Data Flow Diagrams (DFD) |  |
|  | **4.3** | Entity Relationship Diagram (ER-Diagram) |  |
|  | **4.4** | Flow Chart |  |
|  | **4.5** | Website Design Screenshots |  |
| **5** | **DATABASE DESIGN** | |  |
| **6** | **TESTING** | |  |
|  | **6.1** | Type of Testing |  |
|  | **6.2** | Test Cases |  |
| **7** | **CONCLUDING REMARKS** | |  |
|  | **7.1** | Strengths of the System |  |
|  | **7.2** | Limitations |  |
| **8** | **REFERENCES** | |  |
| **9** | **APPENDICES** | |  |
|  | **APPENDIX A** | User Manual |  |
|  | **APPENDIX B** | Source Code Listing |  |
|  | **APPENDIX C** | Input/Output Data Samples |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Title** | **Page No.** |
| 2.1 | Software Process Model |  |
| 2.2 | Incremental model |  |
| 4.1 | Use Case Diagram |  |
| 4.2 | DFD 0 Level |  |
| 4.3 | DFD 1 Level |  |
| 4.4 | DFD 2 Level |  |
| 4.5 | Entity Relationship Diagram |  |
| 4.6 | Login Flow Chart |  |
| 4.7 | Add Lesson Flow Chart |  |
| 4.8 | Payment Status Flow Chart |  |
| 4.9 | studentRegistration.php |  |
| 4.10 | loginorsignup.php |  |
| 4.11 | paymentstatus.php |  |
| 4.12 | Home Page (index.php) |  |
| 4.13 | courses.php |  |
| 4.14 | studentprofile.php |  |
| 4.15 | mycourse.php |  |
| 4.16 | stufeedback.php |  |
| 5.1 | admin (Stores admin detail) |  |
| 5.2 | student (Stores student detail) |  |
| 5.3 | feedback (Stores feedback detail) |  |
| 5.4 | course (Stores course detail) |  |
| 5.5 | lesson (Stores lesson detail) |  |
| 5.6 | couseorder (Stores course order detail) |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page No.** |
| **5.1** | admin (Stores admin detail) |  |
| **5.2** | student (Stores student detail) |  |
| **5.3** | feedback (Stores feedback detail) |  |
| **5.4** | course (Stores course detail) |  |
| **5.5** | lesson (Stores lesson detail) |  |
| **5.6** | couseorder (Stores course order detail) |  |
| **6.1** | Admin login |  |
| **6.2** | User / Student registration |  |
| **6.3** | Add Couse |  |

**1. INTRODUCTION**

**1.1. Company Profile**

At Unified Mentor, we are dedicated to empowering individuals by providing top-tier online certification courses and live training programs. Our mission is to bridge the gap between academic knowledge and industry requirements, equipping learners with the practical skills and expertise needed to excel in today's competitive job market.

**1.1.1. Our Services:**

We offer a diverse range of courses tailored to meet the evolving demands of various industries:

* Data Science
* Full Stack Web Development
* Machine Learning
* Digital Marketing
* UI/UX Design

**1.1.2. Graphic Design Engaging and Effective Training:**

Our training sessions are designed to be interactive, practical, and aligned with industry standards. We emphasize project-based learning, allowing participants to work on live projects that mirror real-world challenges. This hands-on approach ensures that learners not only understand theoretical concepts but also gain the practical experience necessary to excel in their careers.

**1.1.3. Commitment to Social Impact:**

At Unified Mentor, we believe in the transformative power of education. We are committed to supporting individuals from all backgrounds, providing them with the tools and opportunities to achieve their career aspirations. Our focus on practical training and industry-relevant skills aims to create a positive impact by enhancing employability and fostering professional growth among our learners.

**1.2. Introduction To Project**

It is difficult to find time for the training necessary to gain new skills and boost your productivity. With iSchool you’re able to learn at a pace that is comfortable for you. iSchool is a powerful Learning Management System implementing the latest trends in e-learning. E-Learning is learning utilizing electronic technologies to access educational curriculum outside of a traditional classroom. In most cases, it refers to a course, or program delivered completely online. We define eLearning as courses that are specifically delivered via the internet to somewhere other than the classroom where the professor is teaching. E-Learning has been proven to be a successful method of training and education is becoming a way of life for many citizens in India and across the World. iSchool Publisher is a professional team development environment for the rapid development of e-courses by their own.

Any Person who wants to gain new skills can join iSchool. A Person / Student / Learner has to fill up registration form which is absolutely Free. Once Learner registers successfully, they will get UserID / Email and Password for login into Student / Learner Panel. After login they can buy any course as per their choice or requirement which is available in iSchool. They can watch purchased video courses online and can submit their feedback. As well they can update their profile and can change password. Admin of this system will upload new courses which will be available for everyone. Admin can delete or edit student/learner details. Admin can modify course details and can check sells report.

**2. SYSTEM REQUIREMENTS ANALYSIS**

**2.1. Identification of Need**

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained offline at one place only, the ease of service was not there. The information (lectures) was never used to be in a systematic order. It was not possible to provide service for large community from different places at the same time. It was seriously affecting the business. For this reason, we have provided features present system is automated the whole procedure. Present system can be spread to the world so it would be beneficial for the business.

**2.2. Software Requirement Specification**

A software requirements specification is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

**2.2.1. Product perspective**

The software product is a Web Application. The application will be made up of two parts, one administrator who has all the rights and the other user who has limited rights to handle the application. The two users of the system, namely the Teacher / Educator (Admin) and Student / Learner (User) interact with the system in different ways.

**2.2.2. Product Functions**

First of all, it will authenticate the user whether he is Educator (Admin) or Learner (User) the unauthorized person can’t get access to the application.

The admin will be able to Add, delete, and modify Student Details. He can also Add, delete and modify Course and Lesson Details. He can use this application to check report related to sells as well as he can check Payment Status.

The User can edit his own profile and upload his profile picture. He will be able to purchase courses published by admin. User can use application to watch purchased course’s lessons. User can write feedback. Feedback will help Admin to improve the quality of content or service.

**2.2.3. Safety Requirements**

All the data will be saved to database for safety purpose so there will be no data loss. These data can be accessed only by an authorized person so data theft is also not possible in this application.

**2.2.4. Security Requirements**

For preventing unauthorized access to the application, this application has login feature so only granted user can access with defined rights.

**2.3. Data Gathering**

Data collection is the systematic approach to gathering and measuring information from a variety of sources to get a complete and accurate picture of an area of interest. Data collection enables a person or organization to answer relevant questions, evaluate outcomes and make predictions about future probabilities and trends. Accurate data collection is essential to maintaining the integrity of research, making informed business decisions and ensuring quality assurance.

**2.4. Software Process Model**

The Software Process Models are the various processes or methodologies that are being selected for the development of the project depending on the project’s aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out.

The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use.

Choosing right model for developing of the software product or application is very important. Based on the model the development and testing processes are carried out.

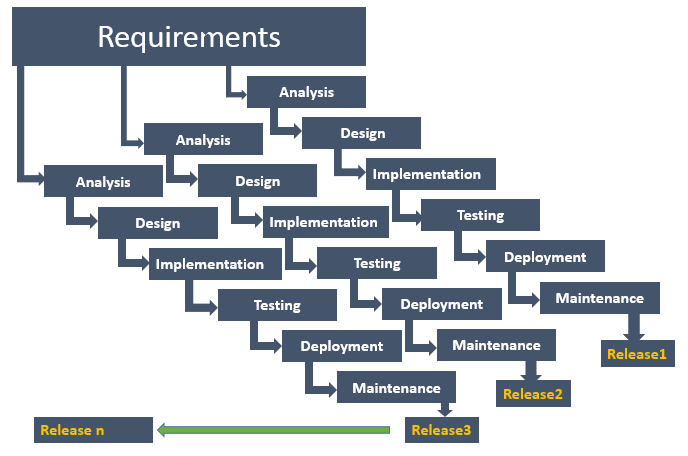
A Process Model describes the sequence of phases for the entire lifetime of a product. Therefore, it is sometimes also called Software Life Cycle. This covers everything from the initial commercial idea until the final de-installation or disassembling of the product after its use.



**Fig. 2.1: Software Process Model**

In order to develop the project “iSchool” we have adopted the Iterative Enhancement Model also known as Incremental Model. This model removes the shortcoming of waterfall model. Since many facts of this system are already known.

It is not a new concept and hence no research is required. A working version can be easily created and hence the system can start working. Rest of the functionalities can be implemented in the next iteration and can be delivered later. As the requirement analysis is also not required. It not being a new technology risk involved is also less. So, one need not perform detailed risk analysis. If redevelopment staff is less than development can be started with a smaller number of people and in next increments others can be involved. As this model combines the advantage of waterfall model and prototyping, clients are always aware of the product being delivered and can always suggest changes and enhancements and can get them implemented. As less amount of customer communication is required one need not apply spiral model in which all types of analysis is done in detail. As the deadline is affordable one need not to for Rapid Application Development model. Iterative enhancement model is useful when less manpower is available for software development and the release deadlines are specified. It is best suited for in house product development, where it is ensured that the user has something to start with. The complete product is divided into releases and the developer delivers the product release by release.



**Fig 2.2. Incremental Model**

**3. FEASIBILITY STUDY**

Feasibility study means to check whether the project is feasible or not, that means possible or not. Some feasibility study regarding this project is as follows: -

**3.1. Economic Feasibility**

The project has shown the economic feasibility by the study of the fact that by using this software the increased number of the users can be given service effectively and efficiently and can save a lot time and saving time means saving money. The cost and benefit analysis has shown that cost that have incurred in developing the project is less than the benefits that the project is going to provide once it is developed, so this project has passed the feasibility test.

**3.2. Technical Feasibility**

Technical feasibility centres on the existing computer system (Hardware, Software etc.) and to what extent it supports the existing system. As the existing system computer system is viable so there is no matter of technical feasibility that is the system is technically feasible. In this type of feasibility study, it is checked whether there is a need of new hardware/software or not. What are the basic requirements of the project? If there is need then how it can be fulfilled. In this context, this project doesn’t need any special hardware or software. It can run on window 7/10 platform. However, Internet and a Web browser is needed to run the web application.

**3.3. Behavioural Feasibility**

The Users are also interested in this project, as it will help them to do work with ease and efficiently without complexity, so they supported the development of this project with full enthusiasm. This shows the behavioural feasibility of the project.

**3.4. Time Feasibility**

It is the determination of whether a proposed project can be implemented fully within stipulated time frame. The project was decided to be done in three months and was thought to be feasible.

**3.5. Operational Feasibility**

In this feasibility study it is determined whether there is need of well qualified operator or simple user. Is there need to train the operator or not? This project is supporting the User-friendly Web application; hence operating this project is so simple. Even a person who has a little knowledge of computer can easily handle this well. There is no need of trained operator.

**4. SYSTEM DESIGN**

The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science in general. System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

**4.1. Use Case**

A use case diagram is essentially a picture showing system behaviour along with the key actors that interact with the system. The use case represents complete functionality. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box are known. Use Case elements are used to make test cases when performing the testing. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed. Use cases can be employed during several stages of software development, such as planning system requirements, validating design and testing software.

Use case diagrams mostly consist of 3 objects: -

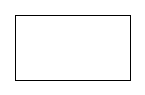
**Actor** - Actor is a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system.



**Use Case** - A Use case represents a function or an action within the system. It’s drawn as an oval and named with the function.

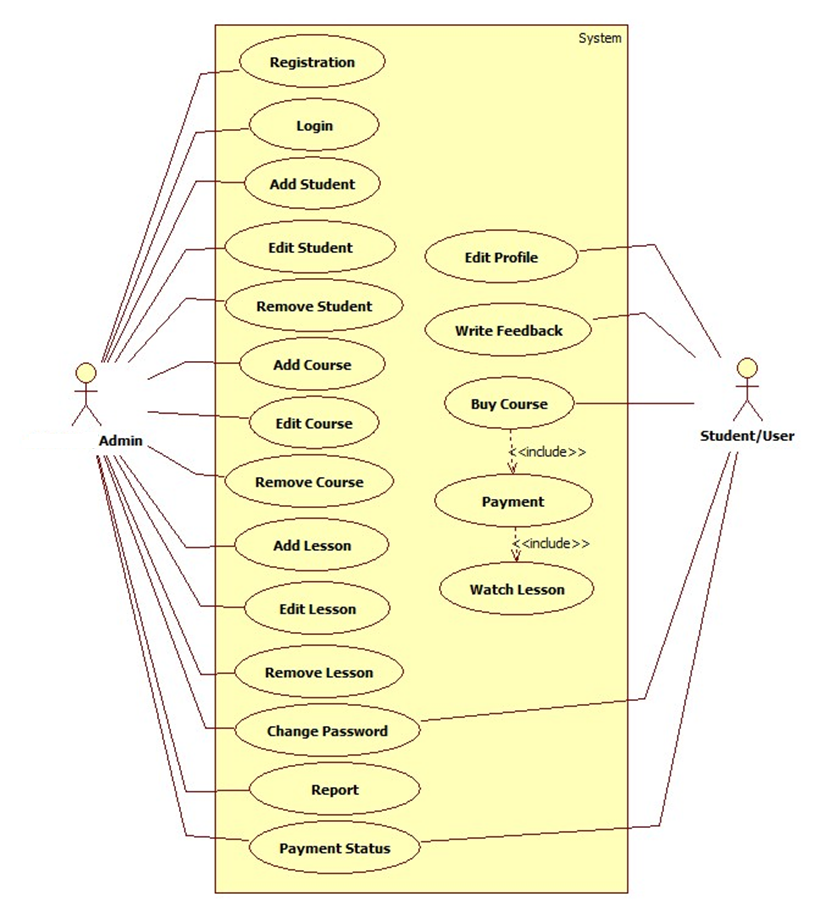


**System** - System is used to define the scope of the use case and drawn as rectangle.



**Include** – This represents required. Symbol of this function is dashed arrow and arrow is labelled with the keyword <<include>>

**Extend** – This represents optional and it is also shown with dashed arrow the arrow is labelled with the keyword <<extend>>



**Fig. 4.1 Use Case Diagram**

**4.2. Data Flow Diagram (DFD)**

Data flow diagram is graphical representation of flow of data in an information system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

**4.2.1 DFD 0 Level**

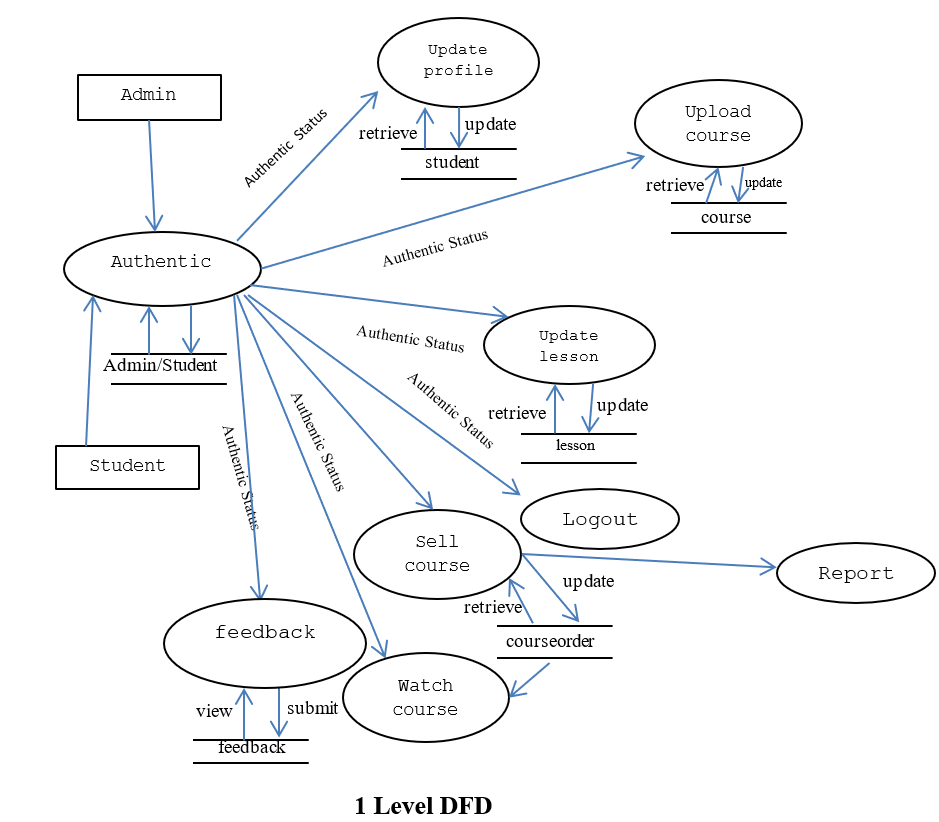
The 0 Level DFD shows flow of data of application. DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analysed or modelled.



**Fig. 4.2: DFD 0 Level**

**4.2.2 DFD 1 Level**

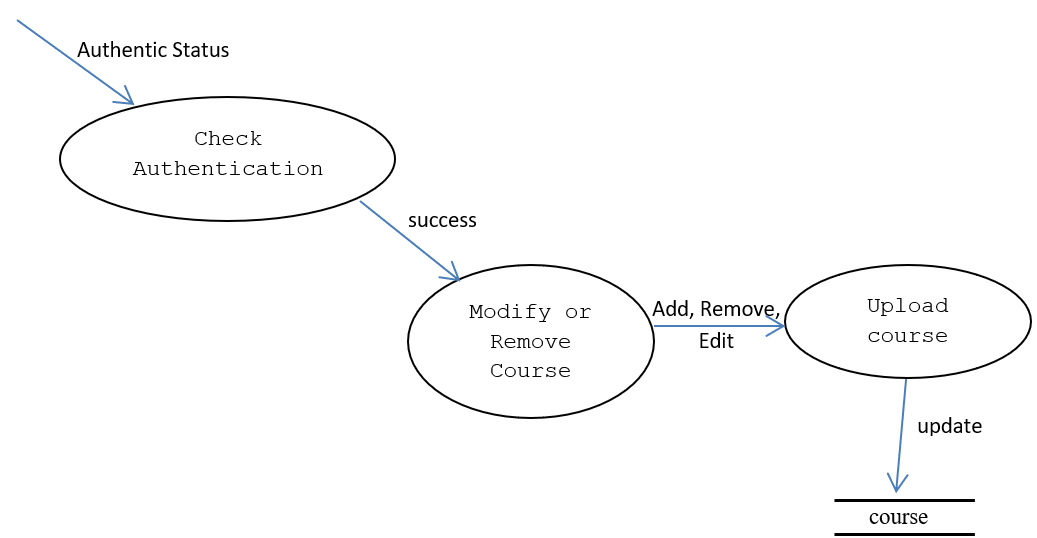
DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. This DFD describes main functions carried out by the system, as we break down the high-level process of the Context Diagram into its sub-processes.

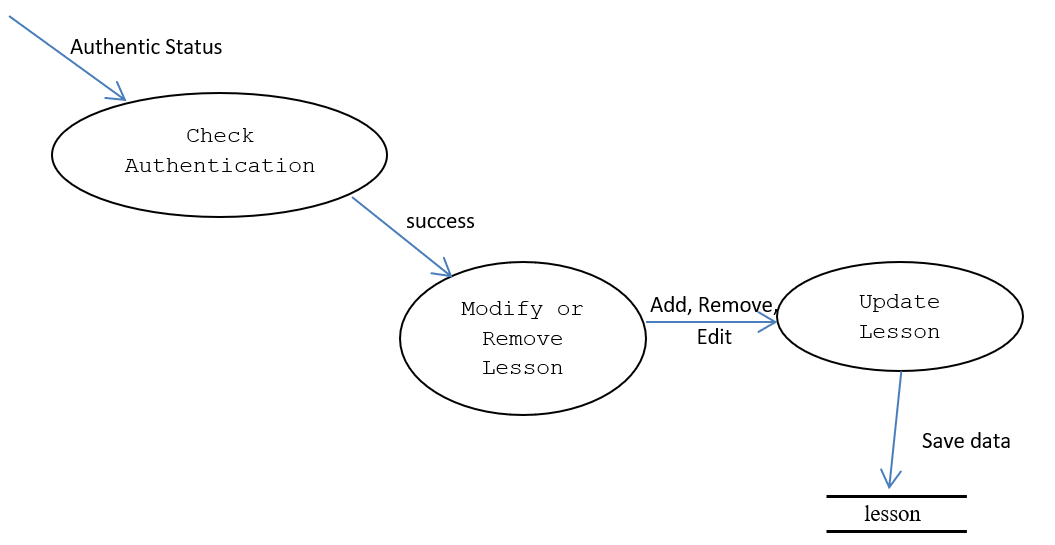


**Fig. 4.3. DFD 1 Level**

**4.2.3 DFD 2 Level**

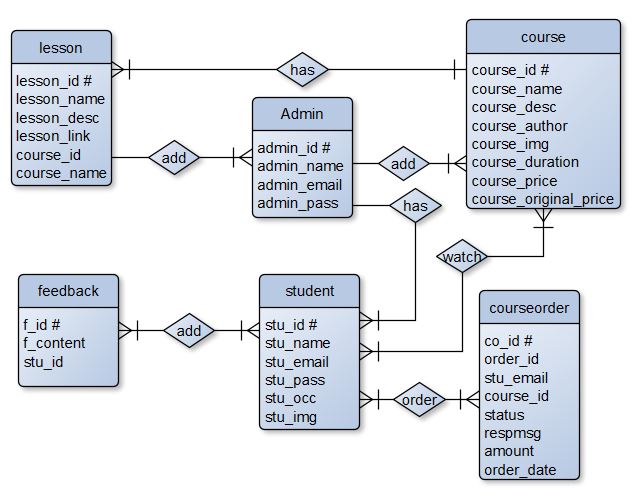
The DFD 2 Level describes flow of data in more detail. DFD Level 2 goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning.





**Fig. 4.4: DFD 2 Level**

**4.3. Entity Relationship Diagram (ER-Diagram)**

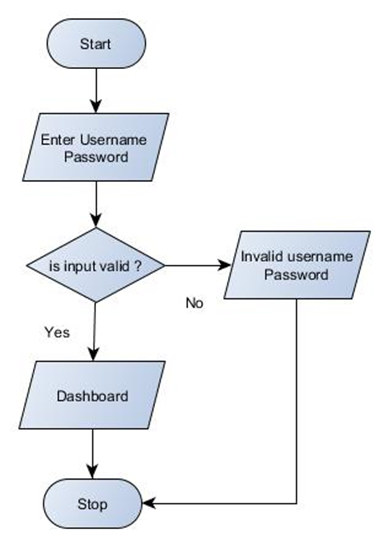
 An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other.

**Fig. 4.5: Entity Relationship Diagram**

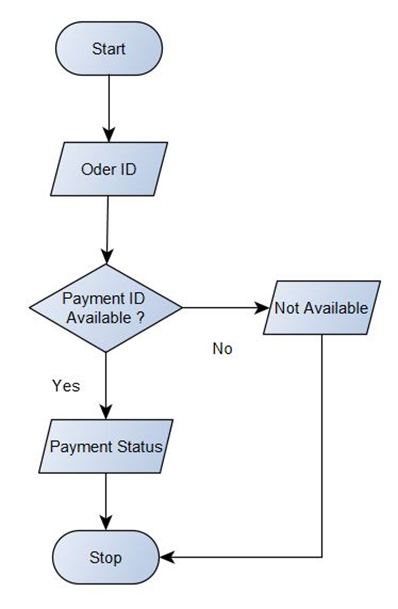
**4.4. Flow Chart**

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

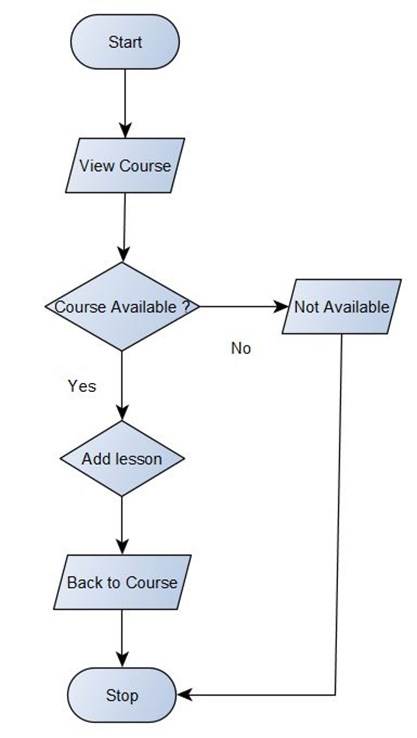
**Fig: 4.6: Login Flow Chart**

****

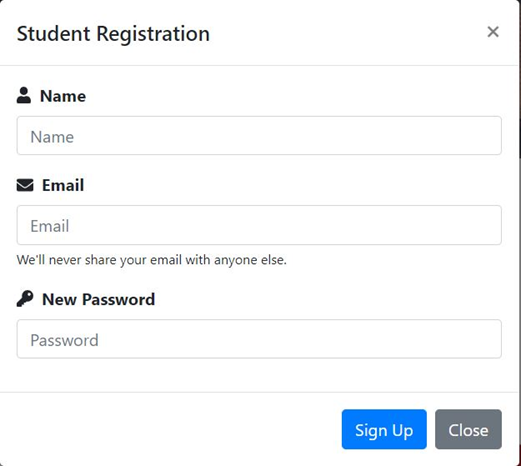
**Fig. 4.8: Payment Status Flow Chart**

****

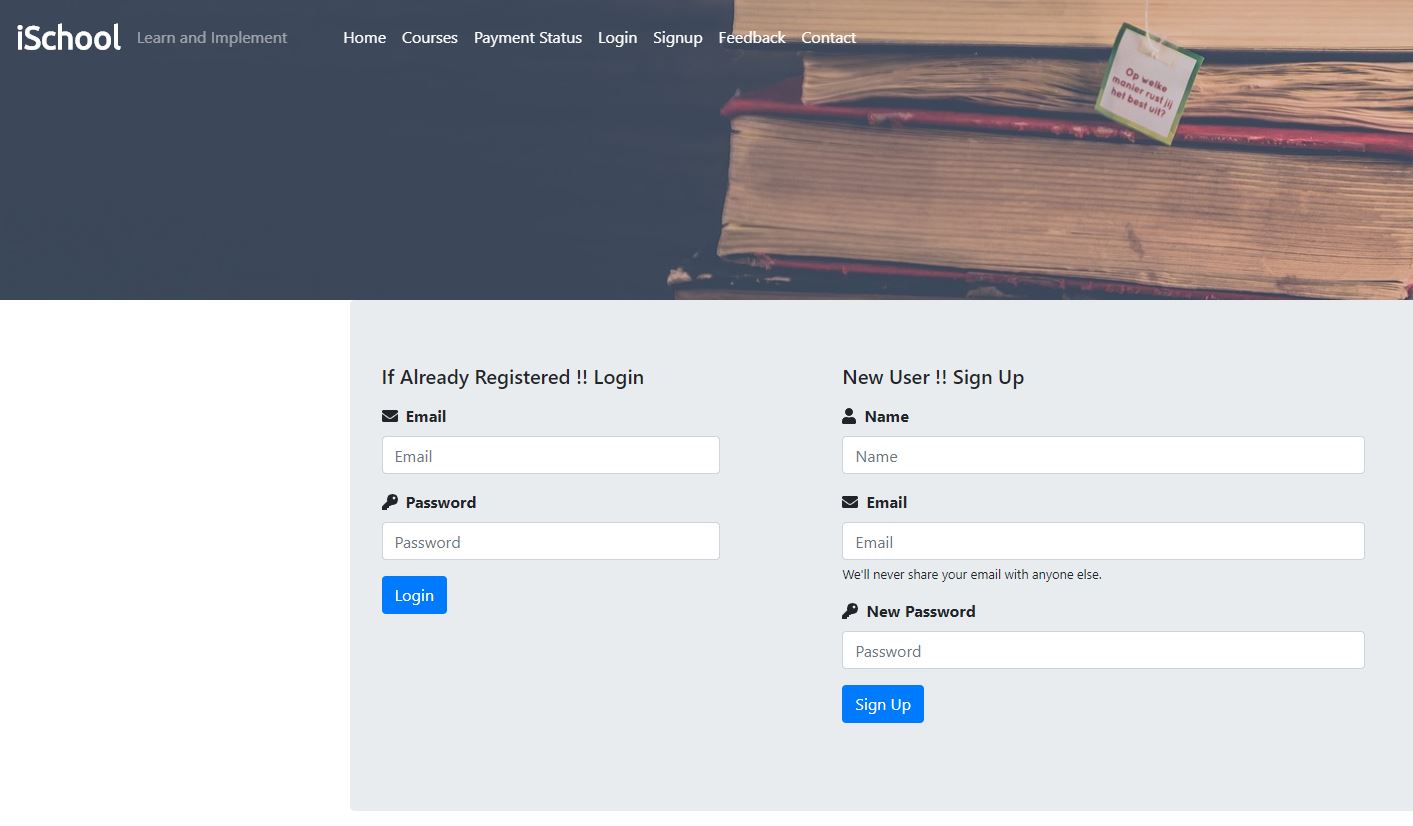
**Fig. 4.7: Add Lesson Flow Chart**

****

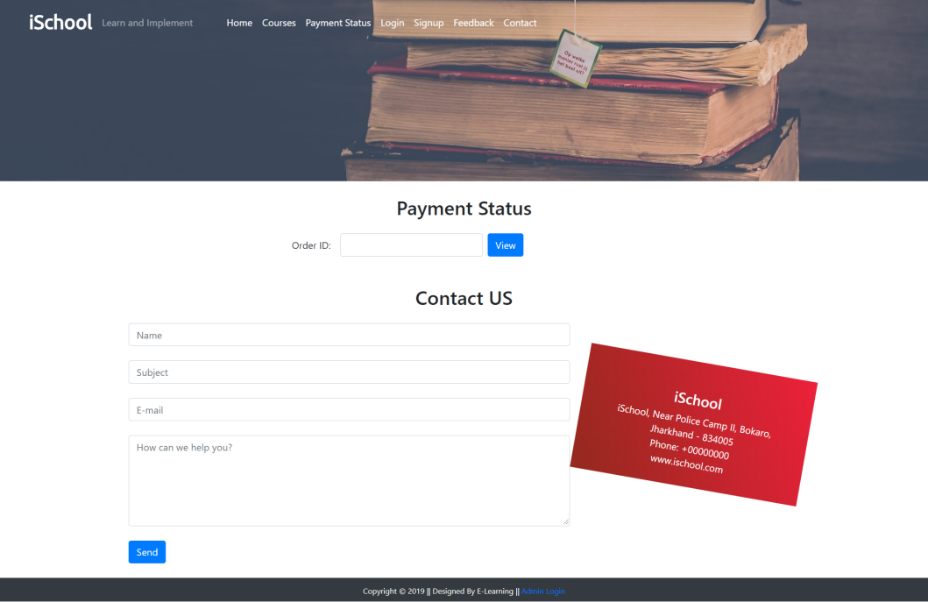
**4.5. Website Design Screenshots**

****

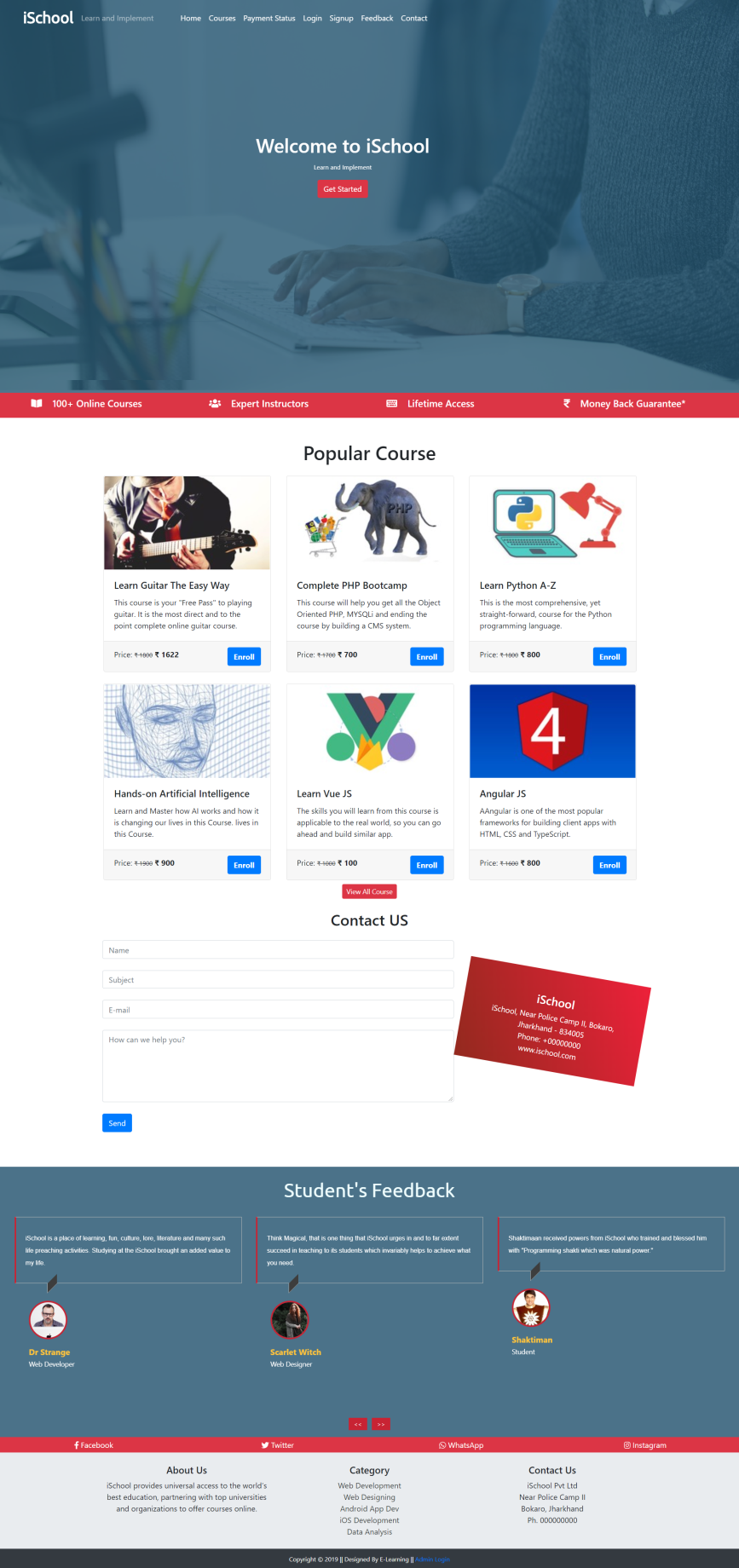
**Fig. 4.9: studentRegistration.php**



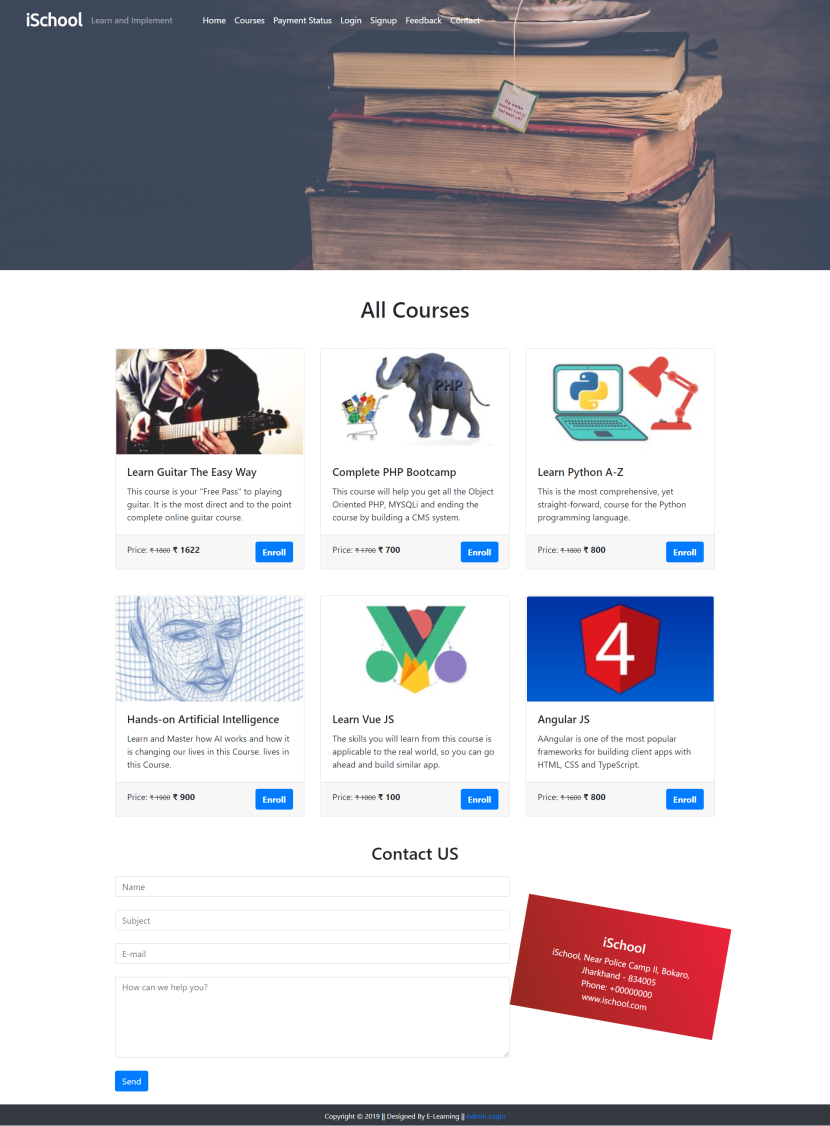
**Fig. 4.10: loginorsignup.php**



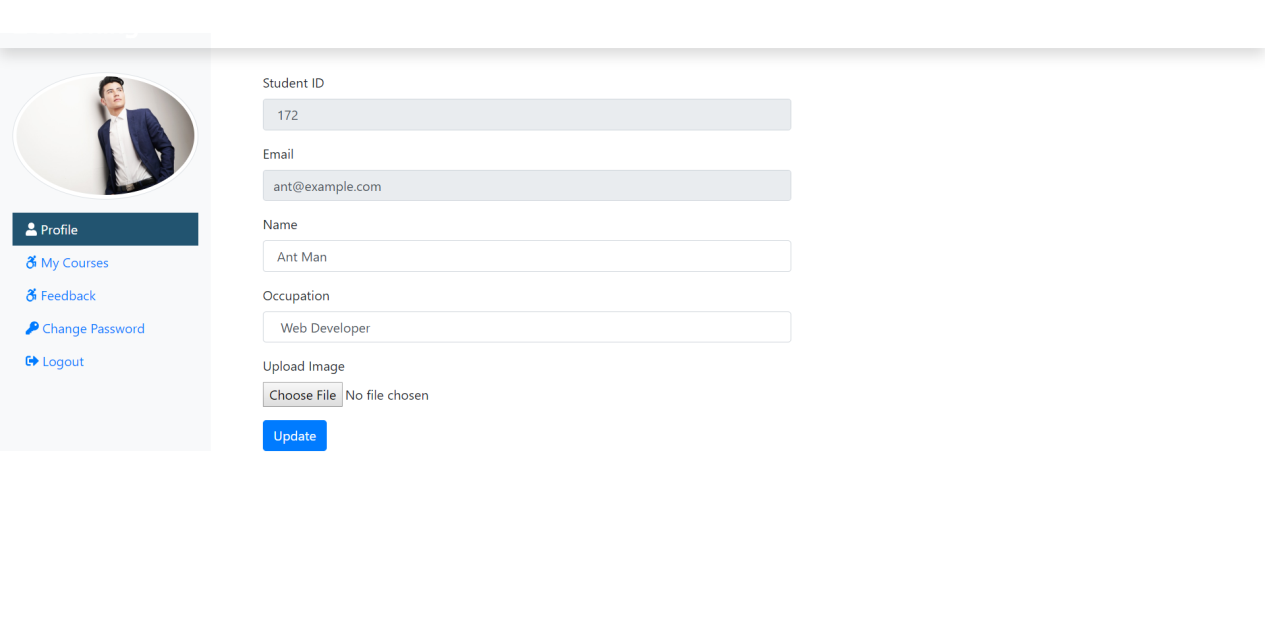
**Fig. 4.11: paymentstatus.php**



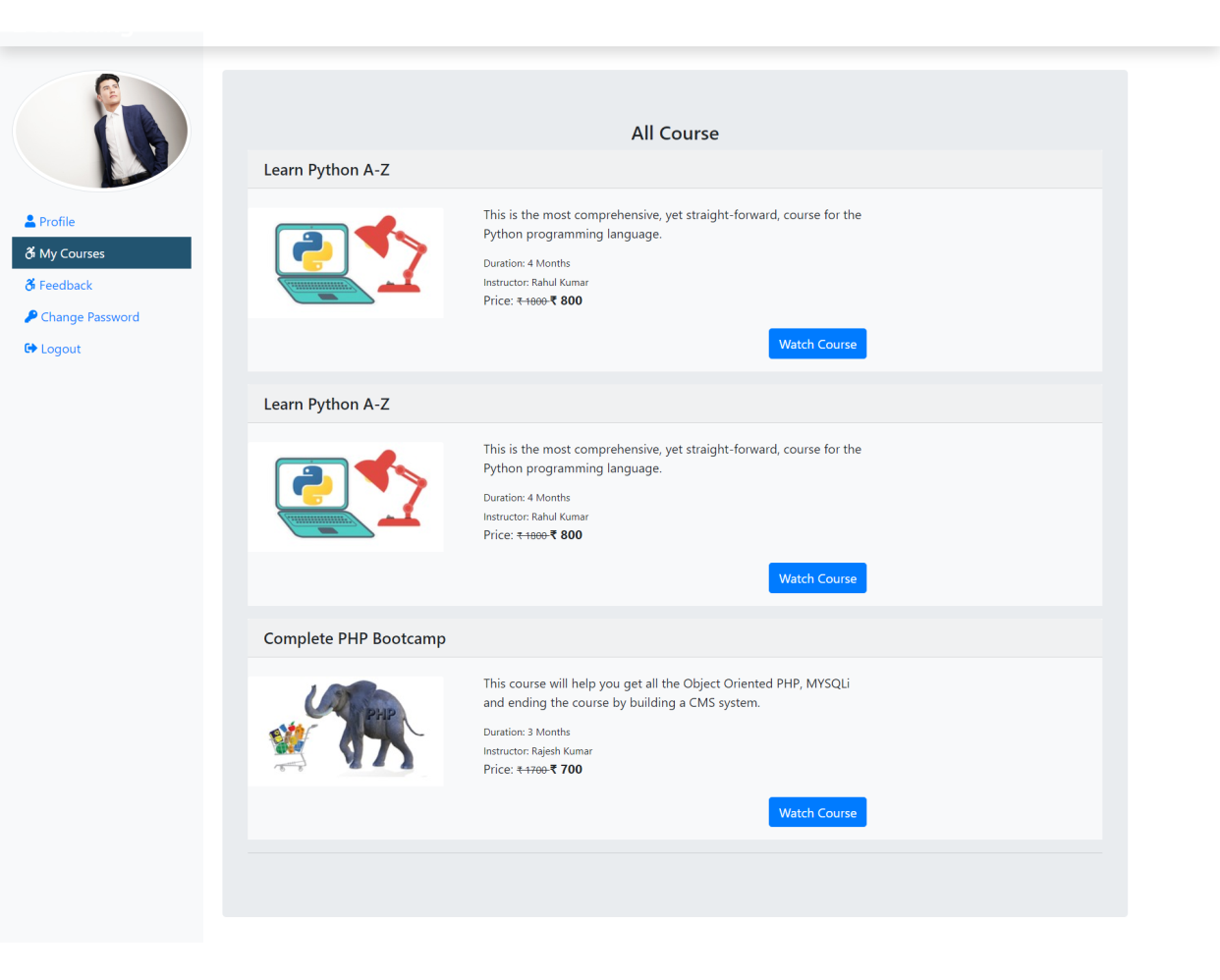
**Fig. 4.12: Home Page (index.php)**



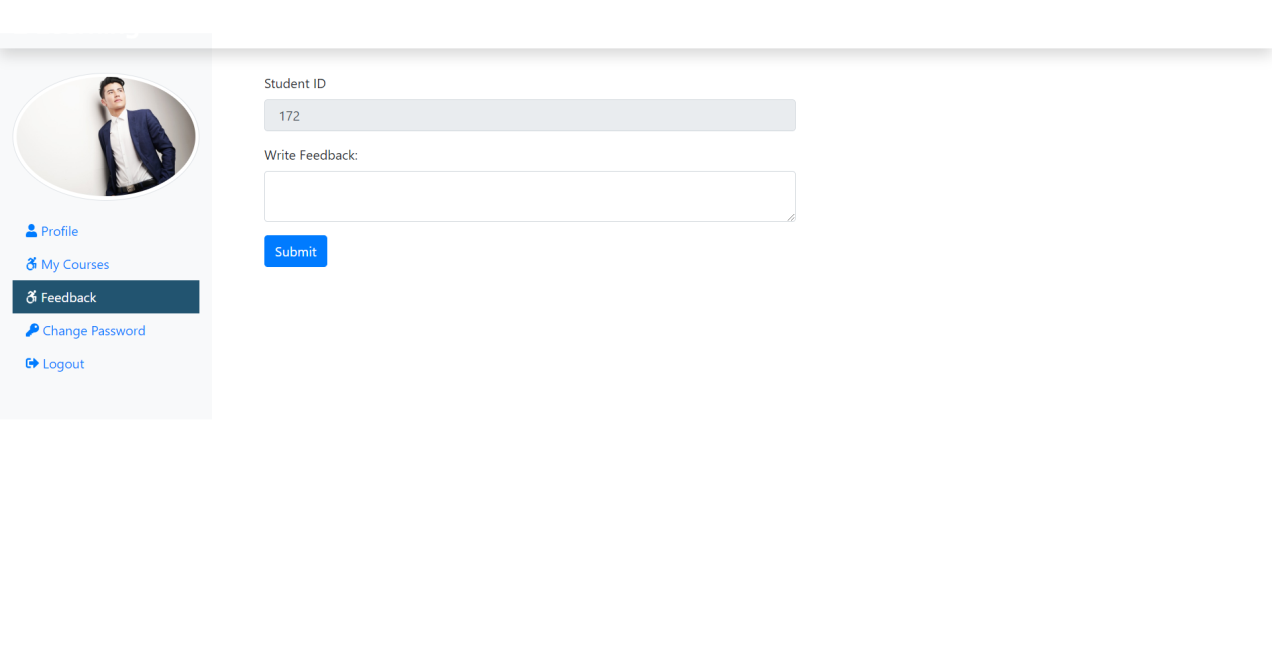
**Fig. 4.13: courses.php**



**Fig. 4.14: studentprofile.php**



**Fig. 4.15: mycourse.php**



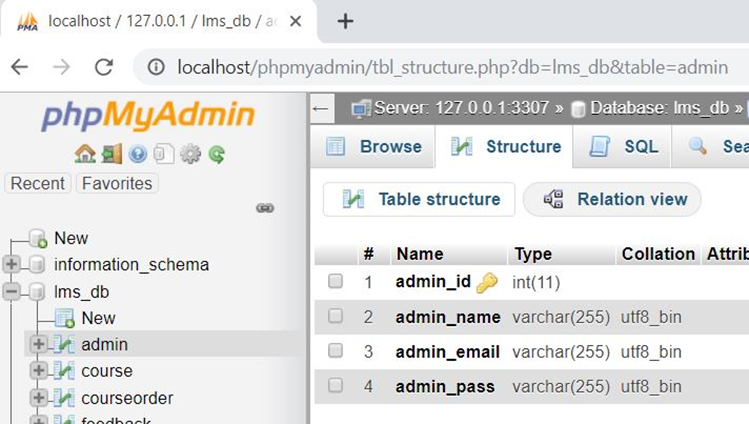
**Fig. 4.16: stufeedback.php**

**5. DATABASE DESIGN**

A data dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management systems keep the data dictionary hidden from users to prevent them from accidentally destroying its contents. For most relational database management systems (RDBMS), the database management system software needs the data dictionary to access the data within a database. For example, the MySQL Database software has to read and write to a MySQL Database. However, it can only do this via the data dictionary created for that particular database.

**Table 5.1: admin (Stores Admin Detail)**

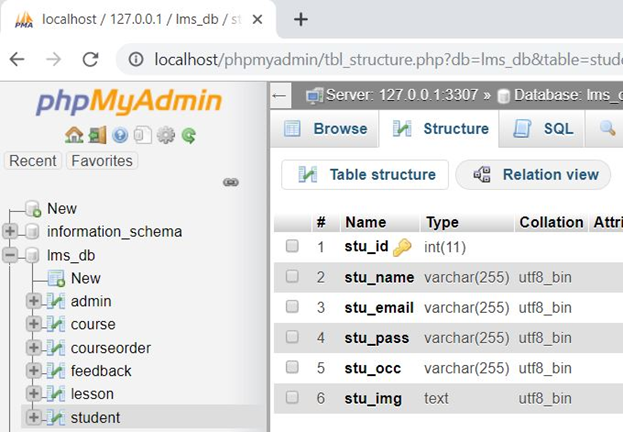
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| admin\_id # | int(11) | Stores Admin ID |
| admin\_name | varchar(255) | Stores Admin Name |
| admin\_email | varchar(255) | Stores Admin Email ID |
| admin\_pass | varchar(255) | Stores Admin Password |



**Fig. 5.1: admin (Stores Admin Detail)**

**Table 5.2: student (Stores Student Detail)**

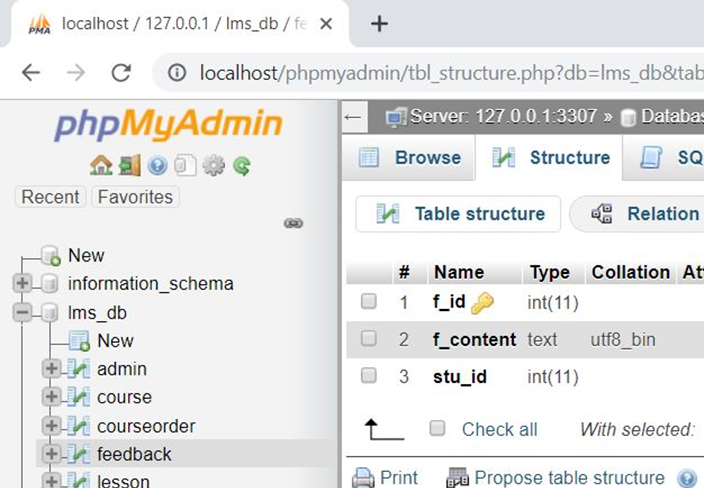
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| stu\_id # | int(11) | Stores student ID |
| stu\_name | varchar(255) | Stores student Name |
| stu\_email | varchar(255) | Stores student Email ID |
| stu\_pass | varchar(255) | Stores student Password |
| stu\_occ | varchar(255) | Stores student occupation |
| stu\_img | text | Stores student profile picture |

****

**Fig 5.2: student (Store Student Detail)**

**Table 5.3: feedback (Stores Feedback Detail)**

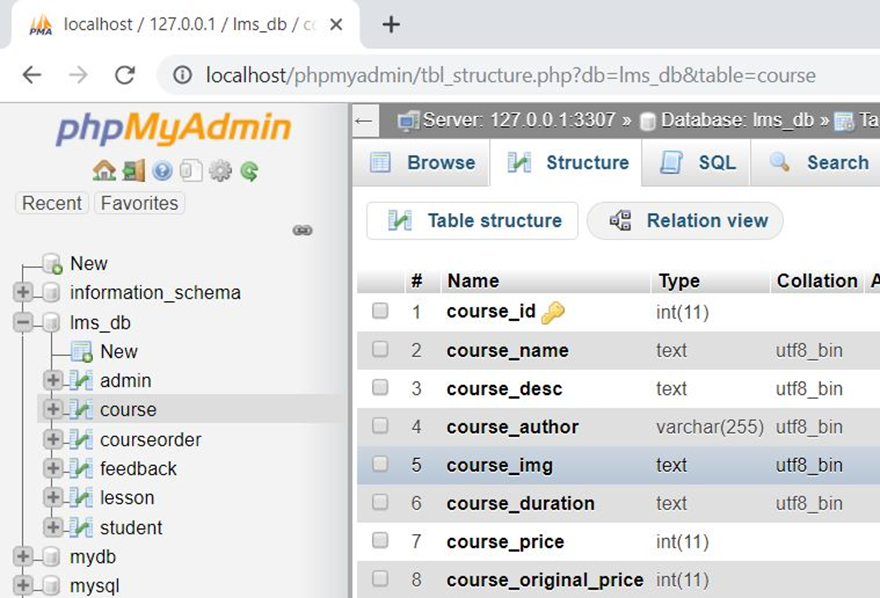
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| f\_id # | int(11) | Stores Feedback ID |
| f\_content | text | Stores Feedback content |
| stu\_id | int(11) | Stores Student ID |

****

**Fig 5.3: feedback (Stores Feedback Detail)**

**Table 5.4: course (Stores Course Detail)**

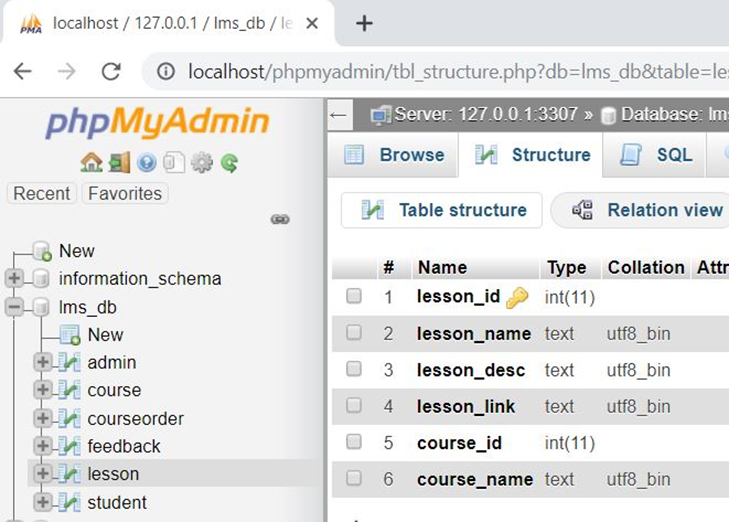
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| course\_id # | int(11) | Stores Course ID |
| course\_name | text | Stores course Name |
| course\_desc | text | Stores course description |
| course\_author | varchar(255) | Stores course author/instructor |
| course\_img | text | Stores course display picture |
| course\_duration | text | Stores course duration |
| course\_price | int(11) | Stores course selling price |
| course\_original\_price | int(11) | Stores course original price |

****

**Fig 5.4: course (Stores Course Detail)**

**Table 5.5: lesson (Stores Lesson Detail)**

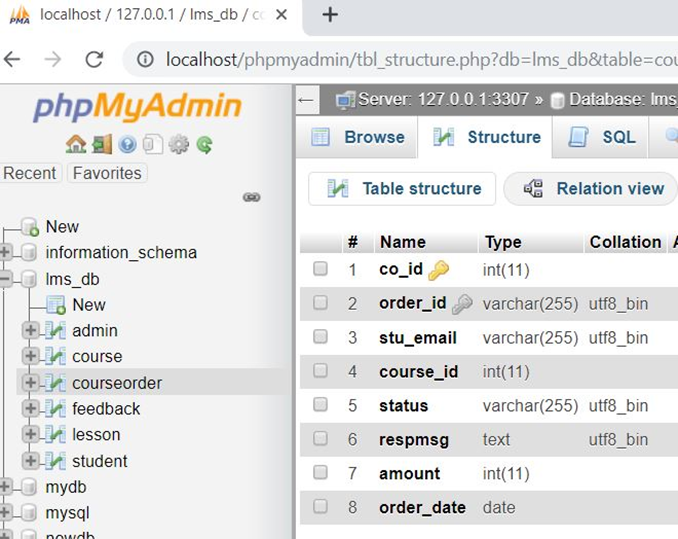
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| lesson\_id # | int(11) | Stores Lesson ID |
| lesson\_name | text | Stores Lesson name |
| lesson\_desc | text | Stores lesson description |
| lesson\_link | text | Stores lesson video link/video file |
| course\_id | int(11) | Stores course ID |
| course\_name | text | Stores course Name |

****

**Fig 5.5: lesson (Stores Lesson Detail)**

**Table 5.6: courseorder (Stores Course order Detail)**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| co\_id # | int(11) | Stores course order ID |
| order\_id | varchar(255) | Stores Order ID (Random) |
| stu\_email | varchar(255) | Stores student email id |
| course\_id | int(11) | Stores course id |
| status | varchar(255) | Stores payment status |
| respmsg | text | Stores payment response msg |
| amount | int(11) | Stores course amount |
| order\_date | date | Stores purchase date |

****

**Fig 5.6: courseorder (Stores Course order Detail)**

**6. TESTING**

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users. In other word software testing is an activity to check that the software system is defect free.

Software testing is primarily a broad process that is composed of several interlinked processes. The primary objective of software testing is to measure software health along with its completeness in terms of core requirements. Software testing involves examining and checking software through different testing processes.

The objectives of these processes can include:

* **Completeness** - Verifying software completeness in regards to functional/business requirements
* **Errors Free** - Identifying technical bugs / errors and ensuring the software is error-free
* **Stability** - Assessing usability, performance, security, localization, compatibility and installation

This phase determines the error in the project. If there is any error then it must be removed before delivery of the project.

**6.1. Type of Testing**

For determining errors various types of test action are performed:

**6.1.1. Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design – the module. Using the detail design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing. The unit test is always white box oriented, and the step can be conducted in parallel for multiple modules.

Unit testing is normally considered an adjunct to the coding step. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins.

**6.1.2. Integration Testing**

A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

**6.1.3. System Testing**

Software is only one element of a larger computer-based system. Ultimately, software is incorporated with other system elements (e.g. new hardware, information), and a series of system integration and validation tests are conducted. Steps taken during software design and testing can greatly improve the probability of successful software integration in the larger system.

A classics system testing problem is “finger pointing”. This occurs when a defect is uncovered, and one system element developer blames another for the problem. The software engineer should anticipate potential interfacing problems and design error handling paths that test all information coming from other elements of the system, conduct a series of tests that simulate bad data or other potential errors at the software interface, record the results or tests to use as “evidence” if finger pointing does occur, participate in the planning and design of system test to ensure that software is adequately tested.

There are many types of system tests that are worthwhile for software-based systems:

**6.1.4. Usability Testing:** Usability Testing is a type of testing done from an end-user’s perspective to determine if the system is easily usable.

**6.1.5. Functionality Testing:** Tests all functionalities of the software against the requirement.

**6.1.6. Performance Testing:** Performance testing is designed to test the run-time performance of software within the context of an integrated system.

**6.1.7. Security Testing:** Security testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration.

**6.1.8. Stress Tests:** Stress tests are designed to confront programs with abnormal situations.

**6.3 Test Case**

A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

**Table 6.1: Admin Login**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Case | Pre-Condition | Test Steps | Test Data | Expected Result | Actual Result | Status  Pass/Fail |
| TC\_Login\_1 | Verify Login | Enter Valid username and valid password | Need a valid username and password to do login | 1. Enter username  2. Enter Password  3. Click Login | Valid username  Valid password | Successful login, Main screen of application should display | Successful login, Main screen of application displayed | Pass |
| TC\_Login\_2 | Verify Login | Enter Valid username and invalid password | Need a valid username and password to do login | 1. Enter username  2. Enter Password  3. Click Login | Valid username  Invalid Password | No Matched Username/ Password | No Matched Username/ Password | Pass |
| TC\_Login\_3 | Verify Login | Enter Invalid username and valid password | Need a valid username and password to do login | 1. Enter username  2. Enter Password  3. Click Login | Invalid username  Valid Password | No Matched Username/ Password | No Matched Username/ Password | Pass |
| TC\_Login\_4 | Verify Login | Enter Invalid username and invalid password | Need a valid username and password to do login | 1. Enter username  2. Enter Password  3. Click Login | Invalid username  Invalid Password | No Matched Username/ Password | No Matched Username/ Password | Pass |

**Table 6.2: User/Student Registration**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Case | Pre-Condition | Test Steps | Test Data | Expected Result | Actual Result | Status  Pass/Fail |
| TC\_SREG\_1 | Verify User Registration Detail | Enter valid name, email, new password | Need valid Data to be entered | 1. Enter name  2. Enter email  3. Enter Password  4. Click Sign up | Valid name, valid email, valid password | Successful, User Added Successfully | Successful, User Added Successfully | Pass |
| TC\_SREG\_2 | Verify Staff Registration Detail | Enter name, already registered email, new password | Need Data to be entered | 1. Enter name  2. Enter Email  3. Enter Password  4. Click Sign up | Valid name, already registered email, valid password | Email ID Already Registered | Email ID Already Registered | Pass |
| TC\_SREG\_3 | Verify Staff Registration Detail | Entering Nothing, Required Fields are blank | - | Click Sign up | Nothing to enter Required fields are blank | Fill required field | Fill required field | Pass |

**Table 6.3: Add Course**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Case | Pre-Condition | Test Steps | Test Data | Expected Result | Actual Result | Status  Pass/Fail |
| TC\_Course\_1 | Verify Course Detail | Enter Valid and correct data | Need valid text and number Data to be entered | 1. Enter Valid Data in appropriate fields  2. Click Submit | Valid Text and Number Data | Successful, Course Added Successfully | Successful, Course Added Successfully | Pass |
| TC\_Course\_2 | Verify Course Detail | Enter invalid and incorrect data | Need text and number Data to be entered | Enter invalid Data in fields | Invalid Text and Number Data | Enter Valid Data | Enter Valid Data | Pass |
| TC\_Course\_3 | Verify Course Detail | Entering Nothing, Required Fields are blank | - | Click Submit | Nothing to enter Required fields are blank | Fill required field | Fill required field | Pass |

**7. CONCLUDING REMARKS**

**7.1 Strengths of the System**

The iSchool E-Learning Management System offers a robust and scalable platform designed to enhance the digital learning experience. Its user-friendly interface makes navigation seamless for learners and administrators alike. The system allows students to register, purchase courses, view lessons, and provide feedback efficiently. Administrators benefit from dynamic features such as course management, student record editing, and sales tracking. The system also ensures data safety and privacy through role-based access and secure database integration. Another major advantage is the system’s accessibility, as it operates entirely online and does not require specialized hardware, making it suitable for widespread adoption across different regions. The use of the Iterative Enhancement Model further adds flexibility for continuous improvement based on user feedback.

**7.2 Limitations**

Despite its strengths, the iSchool system has certain limitations. Currently, the platform lacks real-time communication features such as chat or discussion forums that are often essential in e-learning environments. Additionally, it does not support multilingual content, limiting accessibility for non-English speakers. There is also limited support for quizzes, certification generation, or progress tracking, which are standard in many advanced LMS platforms. While the current security measures are effective, the system would benefit from advanced features like two-factor authentication. Furthermore, it assumes consistent internet access, which might be a challenge in remote areas. Future enhancements could focus on integrating AI-based recommendations, gamification elements, and mobile responsiveness to ensure a more engaging and inclusive learning experience.

**8. REFERENCES**

The following reference has been used to develop the project “iSchool”

**Books:**

* IGNOU Blocks of Systems Analysis and Design
* IGNOU Blocks of Introduction to Software Engineering
* The Complete Reference PHP
* Head First SQL: Your Brain on SQL by Lynn Beighley

**Web Source:**

* [www.google.co.in](http://www.google.co.in)
* [www.wikipedia.org](http://www.wikipedia.org)
* [www.php.net](http://www.php.net)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.getbootstrap.com](http://www.getbootstrap.com)
* [www.fontawesome.com](http://www.fontawesome.com)

**9. APPENDICES**

**Appendix A: User Manual**

The User Manual for the iSchool E-Learning Management System is designed to guide end users—students and administrators—through the system's functionalities. It provides a clear explanation of how to use the application effectively and efficiently.

**For Students / Learners:**

1. **Registration/Login:** New users must fill out a registration form. Once registered, they can log in using their email ID and password.
2. **Course Enrolment:** After logging in, learners can browse available courses. They can purchase courses securely through integrated payment options.
3. **Course Access:** Upon successful purchase, students can view course lessons in video format directly from the dashboard.
4. **Profile Management:** Users can update their personal details and change their passwords from the profile section.
5. **Feedback System:** After completing a course, students are encouraged to provide feedback which helps improve the content quality.

**For Admin:**

1. **Admin Login:** The administrator logs in through a dedicated portal.
2. **Manage Courses:** Admin can add, edit, or delete courses and upload new lessons.
3. **Student Management:** Admin has full access to view, modify, or remove student accounts.
4. **Sales and Reports:** The system displays course sales reports and payment statuses for monitoring business performance.

This user manual ensures that even non-technical users can navigate the platform without difficulty.

**Appendix B: Source Code Listing**

The Source Code Listing for the iSchool project includes all essential scripts, HTML templates, and backend logic necessary to run the web application. The project is developed using HTML, CSS, JavaScript, PHP (or a relevant backend language), and MySQL for database management.

**Main Modules:**

**User Interface (UI):**

* index.html: Home page with navigation links and course previews.
* login.html and register.html: For user authentication.
* dashboard.html: Displays enrolled courses and lessons.

**Backend Scripts:**

* login.php: Handles user authentication and session creation.
* register.php: Processes new user registration and database entry.
* course\_manage.php: Used by admin to add or edit course content.
* fetch\_courses.php: Retrieves course data dynamically for display.

**Database Interaction:**

* MySQL queries for CRUD operations are included in the PHP scripts.
* Prepared statements are used to prevent SQL injection.

**Security:**

* Passwords are encrypted using hashing algorithms before storage.
* Session-based access controls prevent unauthorized entry.

The entire source code is commented for readability, and the codebase is modular for easy maintenance and scalability. This listing serves as a technical reference for developers, testers, and evaluators.

**Appendix C: Input / Output Data Samples**

This section includes representative Input/Output (I/O) Data Samples to illustrate how data is entered into and retrieved from the system. These samples help understand system behaviour during real-time usage.

**1. User Registration (Input)**

* **Name:** Priyanka Marathe
* **Email:** User1@gmail.com
* **Password:** \*\*\*\*\*\*\*\*
* **Occupation:** Student

**Output:**

* Registration Successful
* Auto-generated User ID: U1234
* Redirect to login page

**2. Course Addition by Admin (Input)**

* **Course Title:** "Introduction to Python"
* **Description:** Beginner-level course
* **Duration:** 30 hours
* **Price:** ₹999
* **Author:** Prof. Anjali

**Output:**

* Course added successfully
* Displayed in course list on homepage

**3. Feedback Submission (Input)**

* User ID: U1234
* Course: "Intro to Python"
* Feedback: "Very informative and well-explained."

**Output:**

* Feedback submitted
* Message: "Thank you for your feedback!"

**4. Course Purchase (Input)**

* Course ID: C101
* User Email: User@gmail.com
* Payment Method: UPI

**Output:**

* Payment Success
* Access to course lessons granted
* Invoice generated and emailed