

**KLE SOCIETY'S  
CHIDANAND B. KORE POLYTECHNIC, CHIKODI.  
DEPARTMENT OF TECHNICAL EDUCATION  
BENGALURU - 560 001.**

**2022-23**

*In partial fulfillment for the award of the diploma of*  
**DIPLOMA IN COMPUTER SCIENCE AND  
ENGINEERING**

**A PROJECT REPORT  
ON  
ONLINE MCQ EXAM**

*Submitted by group*

<b>DHARAMRAJ LATTE</b>	<b>339CS20008</b>
<b>SHAHIN MIRAKHAN</b>	<b>339CS20023</b>
<b>VAISHNAVI B THARAKAR</b>	<b>339CS20034</b>
<b>MOHAMMADHUSSAIN SHAIKH</b>	<b>339CS21701</b>

# DEPARTMENT OF TECHNICAL EDUCATION



**KLE SOCIETY'S**

**C. B. KORE POLYTECHNIC, CHIKODI-591201.**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

## **CERTIFICATE**

Certified that this project report entitled "**ONLINE MCQ EXAM**" which is being submitted by **Mr.DHARAMRAJ LATTE** , **Mr.SHAHIN MIRAKHAN** , **Miss.VAISHNAVI B THARAKAR** , **Mr. MOHAMMADHUSSAIN SHAIKH** ,**Reg. No 339CS20008, Reg. No 339CS20023, Reg. No 339CS20034, Reg. No 339CS21701** a bonafide student of **Chidanand B. Kore Polytechnic** in partial full fill meant for the award of **Diploma in Computer Science and Engineering** during the year **2022-23** is record of student's own work carried out under my/our guidance. It is certified that all corrections/suggestions indicated for internal Assessment have been incorporated in the Report and one copy of it being deposited in the polytechnic library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said diploma.

It is further understood that by this certificate the undersigned do not endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the project only for the purpose for which it is submitted.

Prof. AKSHATA.BADIGER  
BILLUR

Guide

Prof S. P. KAPALI

Head of Department

Prof D.D

Principal

Name and signature of Examiner

1. \_\_\_\_\_
2. \_\_\_\_\_

## **ACKNOWLEDGEMENT**

I want to thank my mentor, Mrs.Akshata Badiger, Mr.Sanju Kapali Head of Department of Computer Science and Engineering, from the bottom of my heart for starting an exciting study, for his personal dedication, for engaging discussions, and for her insightful advice. He has consistently supported me throughout the job and provided insightful direction and oversight.

I want to extend my heartfelt gratitude to Prof. Darshankumar Billur, Principal of the K.L.E Society's C. B. Kore Polytechnic, and HOD Mr.Sanju Kapali, Associate Professor Department of Computer Science and Engineering, for your insightful ideas.

I would also want to express my gratitude to the entire Department of Computer Science and Engineering team for their kind cooperation in helping us to successfully finish the seminar. I want to thank all of my friends and classmates for your unwavering assistance and encouragement during the seminar.

I want to use this occasion to thank each and every one of my group members. Without their assistance and contributions, the project would not have been accomplished.

Without acknowledging my adored parents, whose blessings and affection have helped me get to where I am today, the acknowledgements would not be complete. I appreciate the support from each and every member of our family.

## **EXECUTIVE SUMMARY**

An executive summary for an online MCQ test project is a brief overview of the project that highlights its objectives, scope, target audience, features and benefits, and expected outcomes. It provides a snapshot of the project to stakeholders, investors, or decision-makers who may not have the time or expertise to read the entire project proposal.

### **Key Features and Functionality:**

The online mcq exam developed using PHP, MySQL, JavaScript, HTML, CSS, and Bootstrap encompasses the following key features:

**Admin login:** The system allows the administrator to create new categories, add questions and answers, and manage them accordingly. The administrator can view the result user have scored.

**User Login:** The user registers and receives a username and password for easy access. Here the user can view available tests, attend the test and get result.

**Interactive User Interface:** The web-based platform features a user-friendly interface implemented using HTML, CSS, and Bootstrap. It offers an intuitive design, seamless navigation. And responsive layouts for optimal user experience across different devices.

**Secure Data Management:** The system employs MySQL as a database management system to store user data securely. It incorporates robust authentication and encryption mechanisms to protect sensitive information.

### **Benefits:**

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Student can submit online test and get result easily

# CONTENTS

## Chapter 1

1.1 Introduction	1
1.2 Scope of the caption project	2-3

## Chapter 2

2.1 Capstone project planning	
2.1.1 Work breakdown structure	4-5
2.1.2 Timeline developer-schedule	6-9
2.1.3 Cost breakdown structure	10-12
2.1.4 Capstone project risk assessment	12
2.2 Requirements specification	
2.2.1 Functional	13
2.2.2 Non-functional (Quality attributes)	14
2.2.3 User input	15
2.2.4 Technical constraints	16
2.3 Design specifications	
2.3.1 Chosen system design	17
2.3.2 Discussion of alternative design	18-20
2.3.3 Detailed description of components/subsystems	21-22
2.3.4 Components 1-n	22

## Chapter 3

3.1 Approach and methodology	
3.1.1 Discuss the technology	23-24
3.1.2 Methodologies	25

3.1.3 Use cases	26
3.1.4 Programming	27-28
3.1.5 Analysis	28
3.1.6 Process design	29-32
3.1.7 Product design	33-37
3.1.8 Fabrication	38-39

## **Chapter 4**

4.1 Test and validation	
4.1.1 Test plan	40-41
4.1.2 Test approach	42-43
4.1.3 Features tested	44
4.1.4 Features not tested	44
4.1.5 Findings	44-45
4.1.6 Inference	46-47

## **Chapter 5**

5.1 Business aspects	
5.1.1 The market and economic outlook of the project	48
5.1.2 Features	48
5.1.3 How does the project of service fit into the competitive landscape?	49
5.1.4 Who want the possible client or customer?	49-50
5.2 Financial considerations	
5.2.1 Capstone project budget	51
5.2.2 Cost of capstone projection needed for either profit	51
Or nonprofit option	

5.3 Conclusion and recommendations	
5.3.1 State of completion of capstone project	52
5.3.2 Future work	52
<b>Program Code</b>	52-74
<b>References</b>	75
<b>Appendices</b>	76-80

## List of figures

<b>Figure number</b>	<b>Figure name</b>	<b>Page number</b>
Figure 2.3.1.1	System Architecture	17
Figure 2.3.2.1	Data flow diagram level 0	18
Figure 2.3.2.2	Data flow diagram level 1 admin	19
Figure 2.3.2.3	Data flow diagram level 1 user	20
Figure 2.3.3.1	Component diagram	21
Figure 3.1.2.1	Agile methodology diagram	25
Figure 3.1.3.1	Use case	26
Figure 3.1.6.1	List of tables	29
Figure 3.1.6.2	Admin table	29
Figure 3.1.6.3	Category table	30
Figure 3.1.6.4	Courses table	30
Figure 3.1.6.5	Members table	31
Figure 3.1.6.6	Questions table	31
Figure 3.1.6.7	Quiz result table	32
Figure 3.1.7.1	Index page	33
Figure 3.1.7.2	Dashboard	33
Figure 3.1.7.3	Add category	34
Figure 3.1.7.4	View category	34
Figure 3.1.7.5	Add question	35
Figure 3.1.7.6	View question	35
Figure 3.1.7.7	View result	36
Figure 3.1.7.8	Registration page	36
Figure 3.1.7.9	User login page	37
Figure 3.1.7.10	Test page	37



## List of tables

Table number	Table name	Page number
Table 1.2.1	Cost of the project	3
Table 2.1.3.1	Final check	10
Table 2.1.3.2	Labor cost of work	11-12
Table 4.1.2.1	Admin login(test approach)	42
Table 4.1.2.2	Manage test	42
Table 4.1.2.3	Manage question and answer	42
Table 4.1.2.4	User registration	43
Table 4.1.2.5	User login	43
Table 4.1.2.6	Submit test	43
Table 5.2.2.1	Cost (profit/nonprofit)	51

# CHAPTER 1

## 1.1 Introduction

Online MCQ Exam is a technology-driven way to simplify examination activities like defining exam patterns with question banks, defining exam timers, objective/ subjective question sections, and conducting exams using a computer or mobile devices in a paperless manner.

Online MCQ Exam is a cost-effective, scalable way to convert traditional pen and paper-based exams to online and paperless mode. Candidates can appear for the exam using any desktop, laptop, or mobile device with a browser. Exam results can be generated instantly for the objective type of questions

This is developed using HTML, CSS, Bootstrap, JavaScript, PHP and MySQL database. Some changes could be done here to make it more reliable, more automatic and providing more features.

## **1.2 Capstone project Scope Document**

**Capstone project Title:** “Online MCQ Exam”

### **Problem Statement:**

According to my survey, some institutes administer the test manually because there is no online software available. As a result, they spend more time manually preparing the test materials.

### **Objectives:**

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Student can submit online test and get result easily.

### **Capstone project description:**

The goal of this project is to build an "online MCQ exam." Through this project, staff members and administrators can conduct an online exam, answer exam questions, and view the result list. Users must first register before they can log in, take the exam, and receive their results right away. Full stack languages, such as HTML, CSS, JavaScript, PHP, MySQL, and others, will be used to construct this project.

### **Capstone project Deliverables:**

Admin/Staff login

Manage Test

Manage question and answer

View result

User/Student registration

User login

Submit test

## **Key milestones:**

### **Admin Modules**

Admin/Staff login

Manage Test

Manage question and answer

View result

### **User Modules**

User/Student registration

User login

Submit test

## **Constraints:**

Timelines for project is limited.

To acquire knowledgeable and skilled labor.

Cost is limited.

Required limited employees for project.

**Estimated Capstone project Duration:** 72 days (12 Weeks)

**Estimated Capstone project cost:** 43,700

Table 1.2.1

<b>Estimations</b>	<b>Cost(Rupees)</b>
Labor Cost	25,200
Cost of material	8,500
Net Profit	10,000
<b>Total</b>	<b>43,700</b>

## **CHAPTER 2**

### **2.1 Capstone project planning**

#### **2.1.1 Work Breakdown Structure**

In this project there are two main modules: - Admin and User.

##### **Admin modules**

Admin/Staff login: - In this using user name and password admin can login.

Manage test: - In this admin can manage the test (add or delete)

Manage question and answer: - In this admin can add or delete question and answer.

View result: - In this admin can view the result.

##### **User modules**

User/Student registration: - In this user can register with his details.

User login: - using username and password user can login.

Submit test: - In this user can submit the test and get the result

##### **Activities and tasks: -**

###### **Admin/Staff login**

###### **Tasks**

Admin login must design properly. Entered details must validate properly. Entered detail must store in database.

###### **Manage test**

###### **Tasks**

Admin must add or delete test to exam. Added tests must store properly.

## **Manage question and answer**

### **Tasks**

Admin must be able to add and delete question and answer. Added question and answers must store in database.

## **View Result**

### **Tasks**

Admin must manage the result. Result must calculate correctly.

## **User Registration**

### **Tasks**

Student must register with correct details. Registered details must validate. Details must store to database without fail.

## **User Login**

### **Tasks**

User must login with registered details. Entered details must validate. Login details must store to database.

## **Submit test**

### **Tasks**

Submit button must work properly. After submitting result must show properly.

## **2.1.2 Time - line Schedule**

### **Activities & Tasks**

#### **Admin/Staff login (6 Days/42 hours)**

##### Tasks

Admin login must design properly. Entered details must validate properly. Entered detail must store in database.

Collect the Requirement gathering.

Analyse the requirements.

Design Admin login form using HTML, CSS and Bootstrap.

Perform admin login validate using JavaScript.

Perform Database connection using PHP and MySQL.

Testing Admin login to check validation and Database.

This whole task will done by Shahin Khan.

#### **Manage Test (12 Days/84 hours)**

##### Tasks

Admin must add or delete test to exam. Added tests must store properly.

Collect the Requirement gathering.

Analyse the requirements.

Create Manage test page using HTML, CSS and Bootstrap.

Validate Manage test page using JavaScript.

Perform Database connection using PHP and MySQL.

Testing manage test page to check validation and Database.

This whole task will done by Shahin Khan.

### **Manage question and answer (15 Days/105 hours)**

#### Tasks

Admin must be able to add and delete question and answer. Added question and answers must store in database.

Collect the Requirement gathering.

Analyse the requirements.

Create Manage Question and answer page using HTML, CSS and Bootstrap.

Validate Manage Question and answer page using JavaScript.

Perform database connection using PHP and MySQL.

Testing Manage Question and answer page to check validation and database.

This whole task will be done by Vaishnavi Tharakar.

### **View results (7 Days/49 hours)**

#### Tasks

Admin must manage the result. Result must calculate correctly.

Collect the Requirement gathering.

Analyse the requirements.

Design the result page using HTML, CSS and Bootstrap.

Validate Result page using JavaScript.

Perform database connection using PHP and MySQL.

Testing Result page to check validation and database.

This whole task will be done by Vaishnavi Tharakar



### **User/Student registration (6 Days/42 hours)**

#### **Tasks**

Student must register with correct details. Registered details must validate. Details must store to database without fail.

Collect the Requirement gathering.

Analyse the requirements.

Design the registration form for student using HTML, CSS and Bootstrap.

Validate Examination page using JavaScript.

Perform database connection using PHP and MySQL.

Testing registration page to check validation and database.

This whole task will done by Mohammadhussain Shaikh.

### **User login (6 Days/42 hours)**

#### **Tasks**

User must login with registered details. Entered details must validate. Login details must store to database.

Collect the Requirement gathering.

Analyse the requirements.

Design the login page using HTML, CSS and Bootstrap.

Validate login page using JavaScript.

Perform database connection using PHP and MySQL.

Testing login page to check validation and database.

This whole task will done by Mohammadhussain Shaikh.

**Submit test** (8 Days/56 hours)

Tasks

Submit button must work properly. After submitting result must show properly.

Collect the Requirement gathering.

Analyse the requirements.

Design the submit button using HTML, CSS and Bootstrap.

Validate submit button using JavaScript.

Perform database connection using PHP and database.

Testing submit button to check validation and database.

This whole task will done by Dharmaraj Latte.

### 2.1.3 Cost Breakdown Structure

#### Analyse your Work Breakdown Structure

Admin/Staff login

Manage Test

Manage question and answer

View result

User/Student registration

User login

Submit test

#### Estimate the cost of materials

The cloud server cost is of 5000.

PHP Designer license key of rupees 3450.

XAMPP Severe is of free Source.

#### Overhead costs

If we need extra cost for live server.

#### Build contingency into your CBS

As per the estimated cost we need deliver the project.

#### Final-check

As per the client budget we have to provide the project budgets.

Table 2.1.3.1

<b>Title</b>	<b>Cost(Rupees)</b>
Labour Cost	25,200
Cost of material	8,500
Net Profit	10,000
<b>Estimated Cost</b>	<b>43,700</b>

### Estimate the labor cost of work

Table 2.1.3.2

Modules	Tasks	Time			Total Cost (hour*amount per hour) (Rupees )
		Hours per task	Hours	Amount per hour (Rupees )	
Admin/Staff login	Admin must login with correct details.	21	42	60	42*60=2,520
	Entered details must validate properly.	17			
	Entered detail must store in database.	14			
Manage Test	Admin must add or delete test to exam	44	84	60	84*60=5,040
	Added tests must store properly.	40			
Manage question and answer	Admin must able to add and delete question and answer.	56	105	60	105*60=6,300
	Added question and answers must store in database	49			
View result	Admin must able to see the result	28	49	60	49*60=2,940
	Result must calculate correctly	21			
User/Student registration	User must register with all details	21	42	60	42*60=2,520
	Registered details must validate	7			
	Details must store to database without fail	14			
User login	User must login with registered details	21	42	60	42*60=2,520
	Entered details must	7			

	validate				
	Login details must store to database	14			
Submit test	Submit button must work properly	21	56	60	56*60=3,3 60
	After submitting result must show properly	35			

### 2.1.4 Risk assessment

Admin login form must work properly without any errors or issues.

Admin login page must visible properly with all the design.

Admin must be able to access the exam conducting page.

Admin must be able to see the list of tests.

All questions and answers must store to database properly.

Correct answers must validate properly.

Admin must be able to see the result.

User registration form must design properly.

User details must validate properly without any issues.

User login page response properly.

User login details must validate correctly.

Submit button must work properly.

After submitting the test website must show result properly.

## **2.2 Requirements specifications**

### **2.2.1 Functional requirements:**

#### **Admin Aspect**

Logging into the system

Accepting registration of candidates

Adding/editing/deleting the question

Creating questions

Posting question

Posting multiple option to respective question

Giving correct answer

Time limit

Set marks

Negative marks if required

#### **User / Student:**

Requesting registration

Logging into the system

Selecting the questing

Appearing for the examination

Reviewing the given response

## **2.2.2 Non-functional requirements**

### **Usability**

Usability is a quality attribute used to access how easy the interface is to use. Usability is ease of use. It tells how user friendly the interface is. It includes memorability, learnability, and satisfaction. Our software interface has all the above quality. Any kind of user can easily understand the interface.

### **Reliability**

Reliability is how much the system is consistent in different platforms. The ability of an apparatus, system to consistently perform its required function, on demand and without degradation or failure.

### **Integrity**

Integrity means doing the right thing in a reliable way. Data integrity is a fundamental component of security. In its broadcast use, “Data Integrity” refers to the accuracy and consistency of data stored in a database, data mart or another construct. Data integrity is the overall completeness, accuracy and consistency of data.

### **Performance**

Performance is also a major non-functional requirement. Performance Requirements about resources required, response time, transaction rate or anything else having to do with performance.

### **2.2.3 User inputs**

#### **Admin:**

Username and password

Tests

Questions and correct answers

Allow users to login

#### **User / Student:**

Full name, last name, username, password, contact number, email, address,  
select year, select course, select semester.

Username and password

Correct answers



## **2.2.4 Technical constraints**

### **Requirements for developing:-**

Average I3 processor.

Average 4GB RAM.

Average 512GB hard disk.

### **Hardware requirement for deployment:-**

Average I3 processor.

Average 2GB RAM.

Average 256GB storage.

### **Software requirement for development:-**

OS (Window's, MAC, Linux).

XAMP Server.

VS Code.

Chrome.

### **Software requirement for deployment:-**

OS.

Chrome.

### **Language used or technology:-**

Content language (client side language) (HTML, CSS, Bootstrap, JavaScript).

Backend (server side language) (PHP).

Database (MySQL).

## 2.3 Design Specification

### 2.3.1 Chosen system design

System architecture diagrams provide a visual illustration of a system's various components and show how they communicate and interact with each other. These diagrams document a system's structure and architecture.

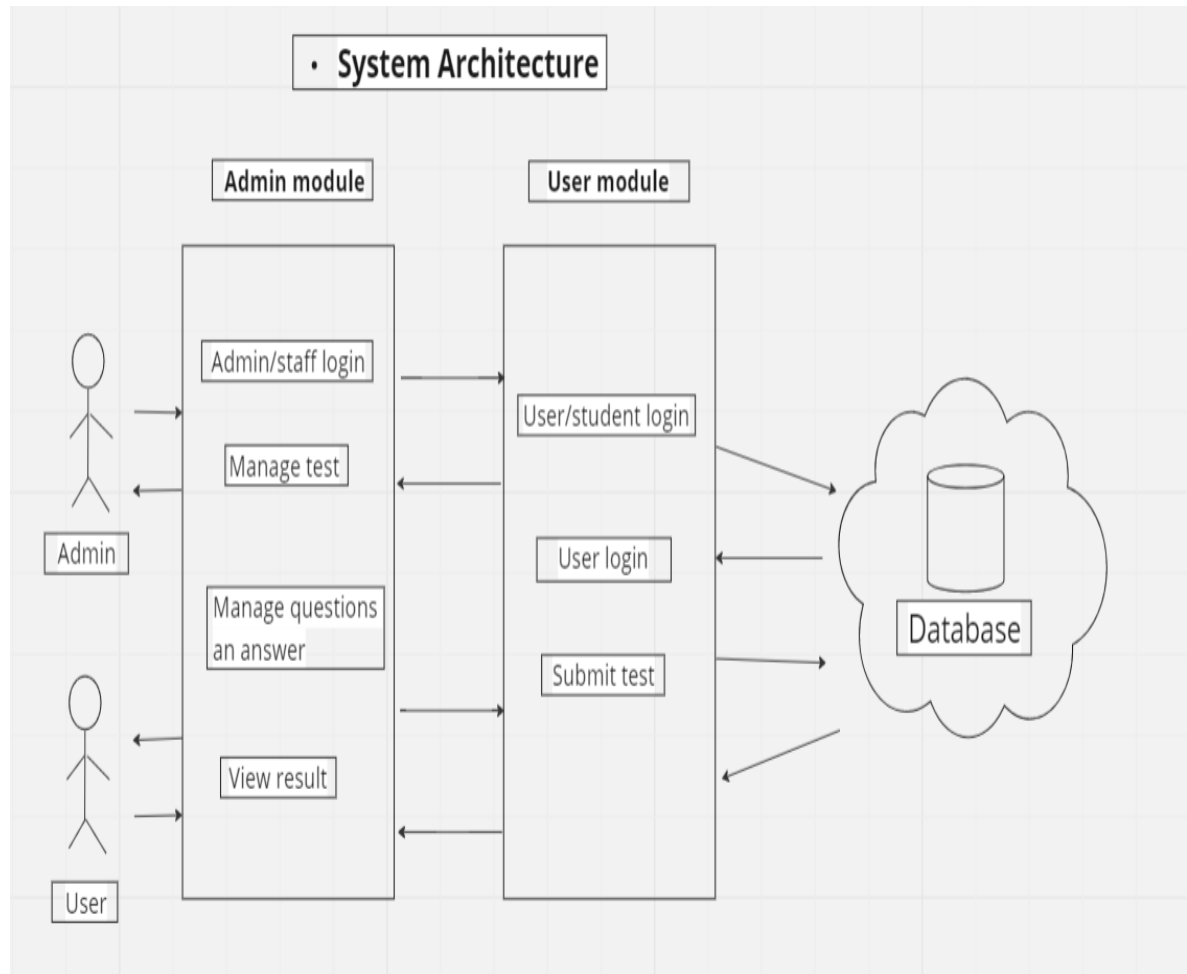


Figure 2.3.1.1

### 2.3.2 Discussion of alternative designs

A data flow diagram (DFD) maps out the flow of information for any process or 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.

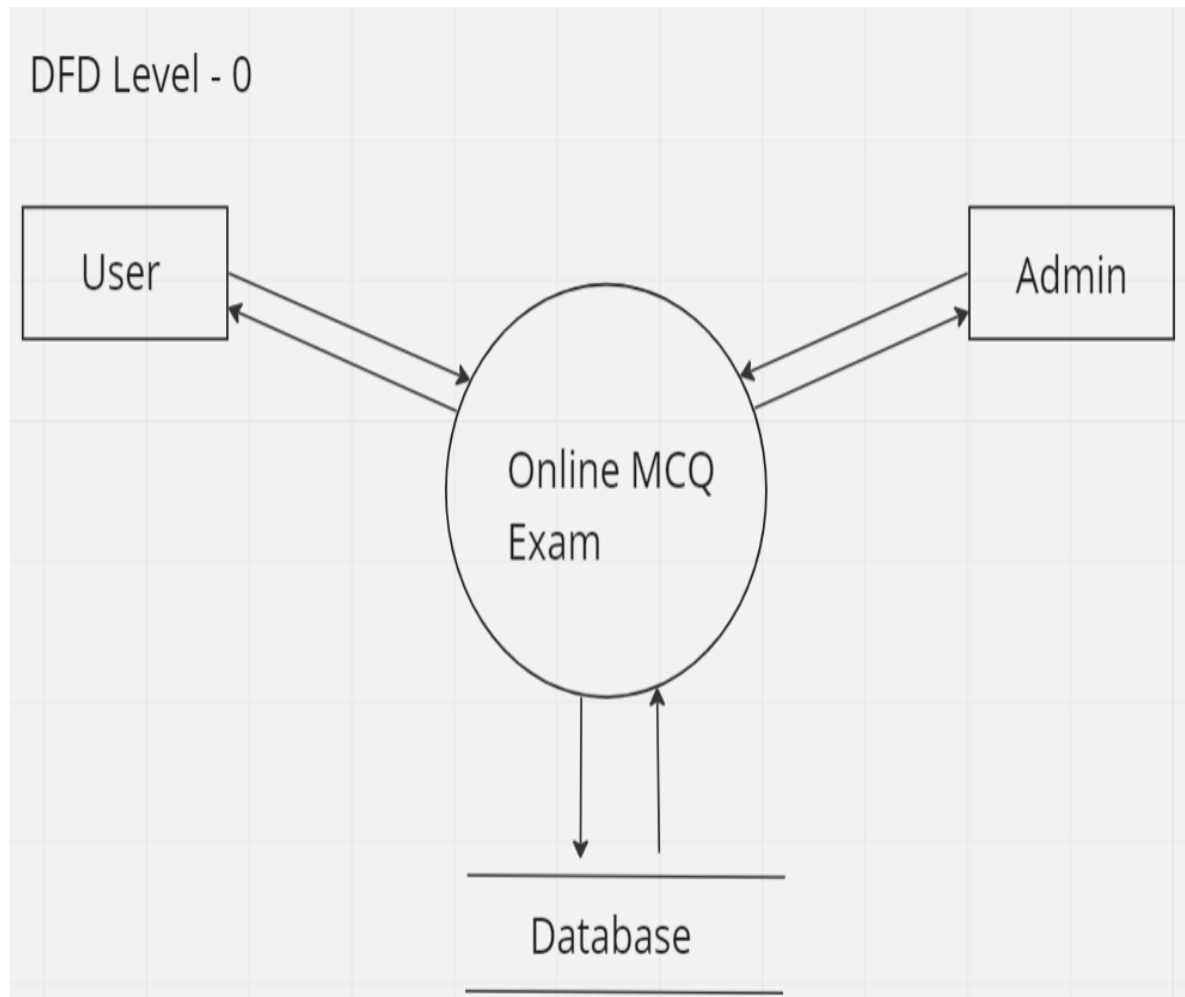


Figure 2.3.2.1

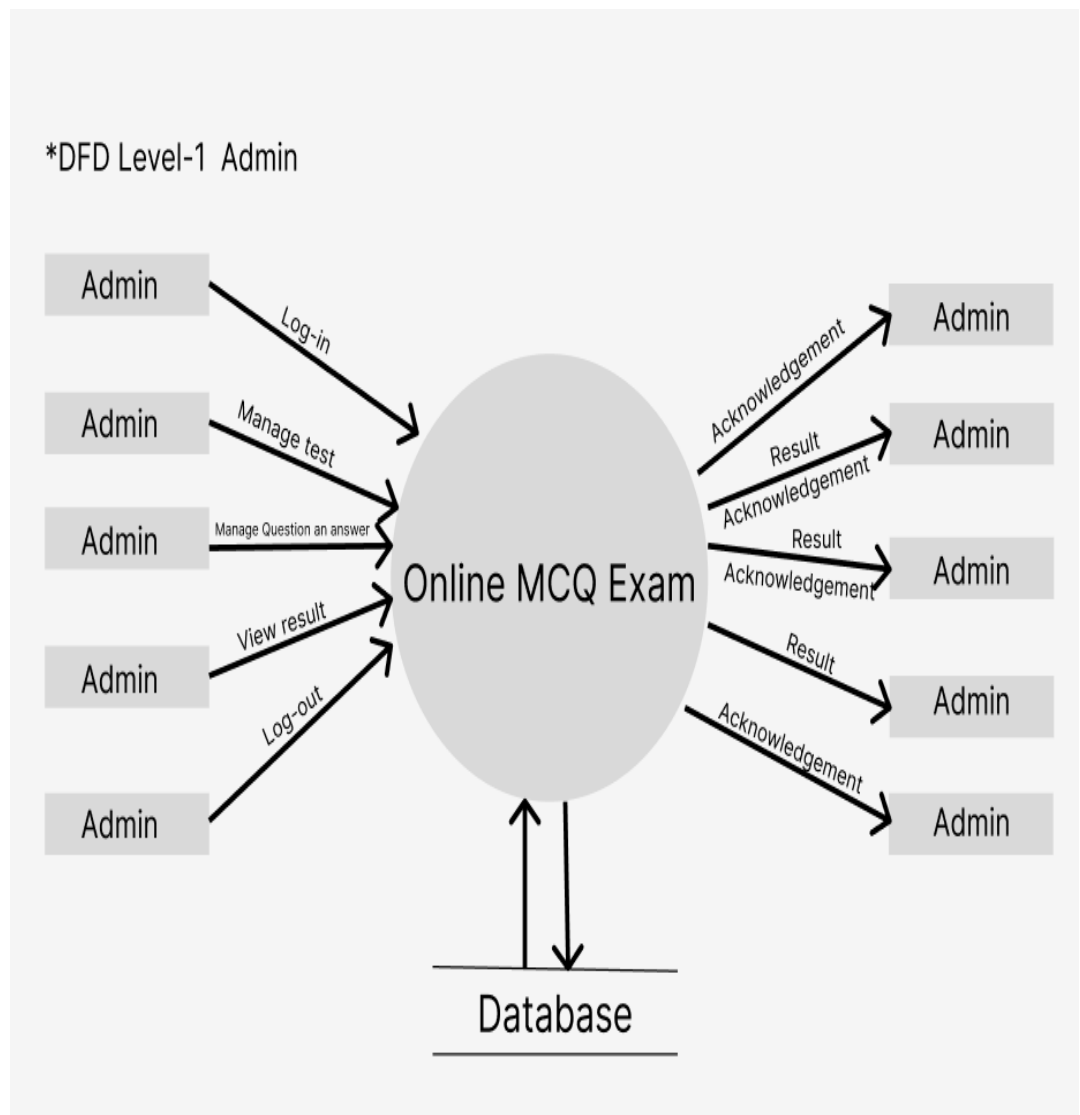


Figure 2.3.2.2

\*DFD Level-1 User

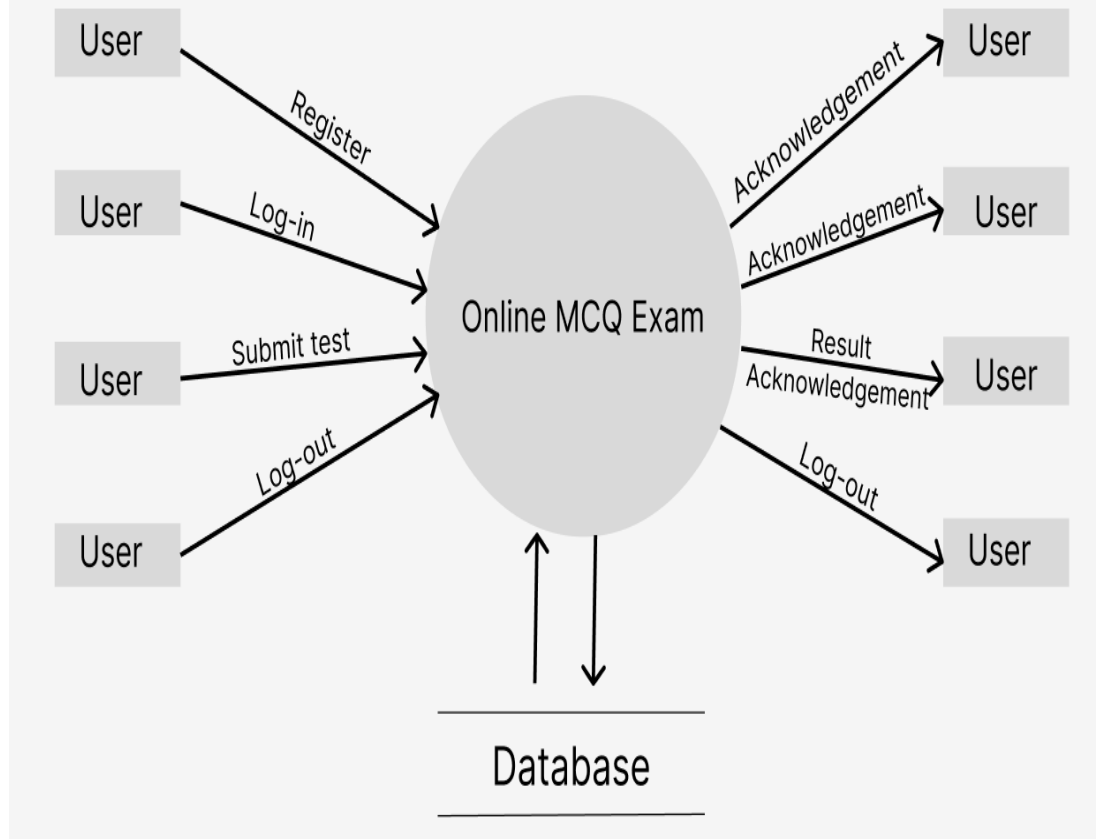


Figure 2.3.2.3

### 2.3.3 Detailed description of components/Subsystems

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

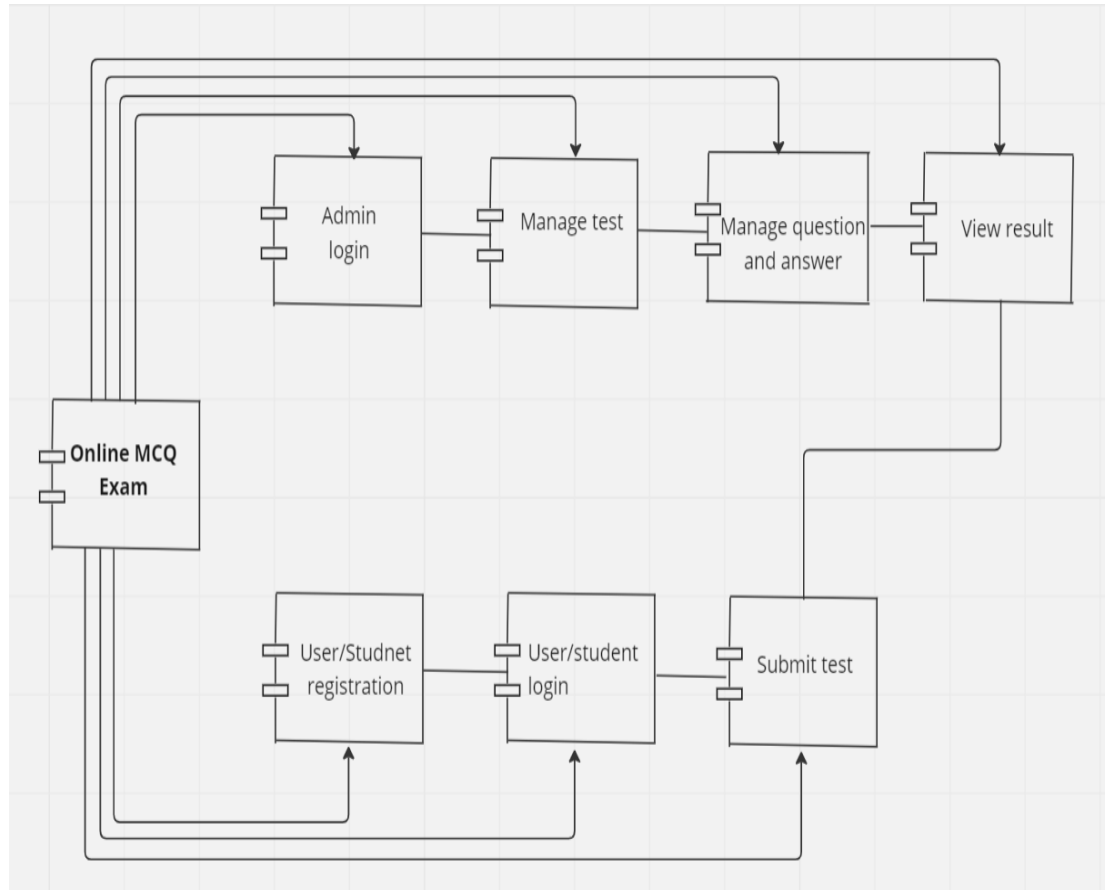


Figure 2.3.3.1

#### Admin/Staff login

Here admin can login to the form using admin username and password. After logging in he can add the tests to conduct online exam, he can add or delete questions and answers to the particular test. He can allow user who wants to login to the form and also he can manage the result.

#### Manage Test

After admin logged in to the form admin can easily manage the tests he had added. He can add the tests and he can also update the test.

**Manage question and answer**

Admin can also manage the question and answers. It means admin can add or delete the questions to the particular test they want and also they can choose the correct answer

**View result**

Admin can view the result list of user have scored. They have access to see the result list.

**User/Student registration**

User can register himself by adding his information. User have to add detailed information with username and password, so he can easily login to the form.

**User login**

After registration user can login to the form using username and password which he had entered while registration. After logging in he is able to attend the test.

**Submit test**

After login user can enter into the test, and he can attend the test. After clearing all questions he can submit the test and get result quickly without any issues.

**2.3.4 Components 1-n**

Admin login

Manage test

Manage question and answer

View result

User/student registration

User/student login

Submit test

## **CHAPTER 3**

### **3.1 Approach and Methodology**

#### **3.1.1 Discuss the technology**

##### **Web technology**

A place connected to the internet, where a company, organization, etc. Puts information that can be found on the World Wide Web.

##### **Types of web technology: -**

Browsers

HTML and CSS

Programming Languages

Frameworks

Web Servers

Databases

Protocols

Lastly, data Formats

##### **Advantages**

We can access from any location.

No data loss.

Data can be recovered.

Be available every time.

We can save time.

##### **Cloud based technology**

To live the project we need to purchase cloud sever.

Cloud based technology is the use of software and services via the internet .These applications commonly include data storage, networking, servers and databases. User can access their cloud hosted tools with any device that is connected to the internet.



## **Types of cloud based technology**

SAAS: - Software as a Service

PAAS: - Platform as a Service

IAAS: - Infrastructure as a Service

## **Advantages**

Usability and accessibility

Security

Cost efficient

Convenient sharing of files

Automation

## **Open-Source web technology**

For our project we not require to purchase any software's and libraries.

## **XAMPP**

XAMPP is a cross-platform and open source tool, which makes it an ideal choice of web developers. It is the acronym of X-cross platform, Apache, MySQL, PHP, and Perl.

## **PHP**

PHP is an open source scripting language used for creating dynamic and interactive web pages and various digital platforms.

## **PhpMyAdmin**

PhpMyAdmin is an open source and free administration tool for MySQL

## **Advantages**

Community-Driven Reliability

Community-Driven Security

Low Cost on an Ongoing Basis

Better, Community-Based Collaboration

### 3.1.2 Methodologies

Agile methodology is a process for managing a project that involves constant collaboration and working in iterations. Agile project management works off the basis that a project can be continuously improved upon throughout its life cycle, with changes being made quickly and responsively. Agile is one of the most popular approach to project Management due to it's flexible, adaptability to change, and high level of customer input.

#### Phases of Agile model

Requirement Gathering  
Design the requirements  
Construction or Iterations  
Testing and Quality assurance  
Deployment  
Feedback



Figure 3.1.2.1

### 3.1.3 Use Cases

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen.

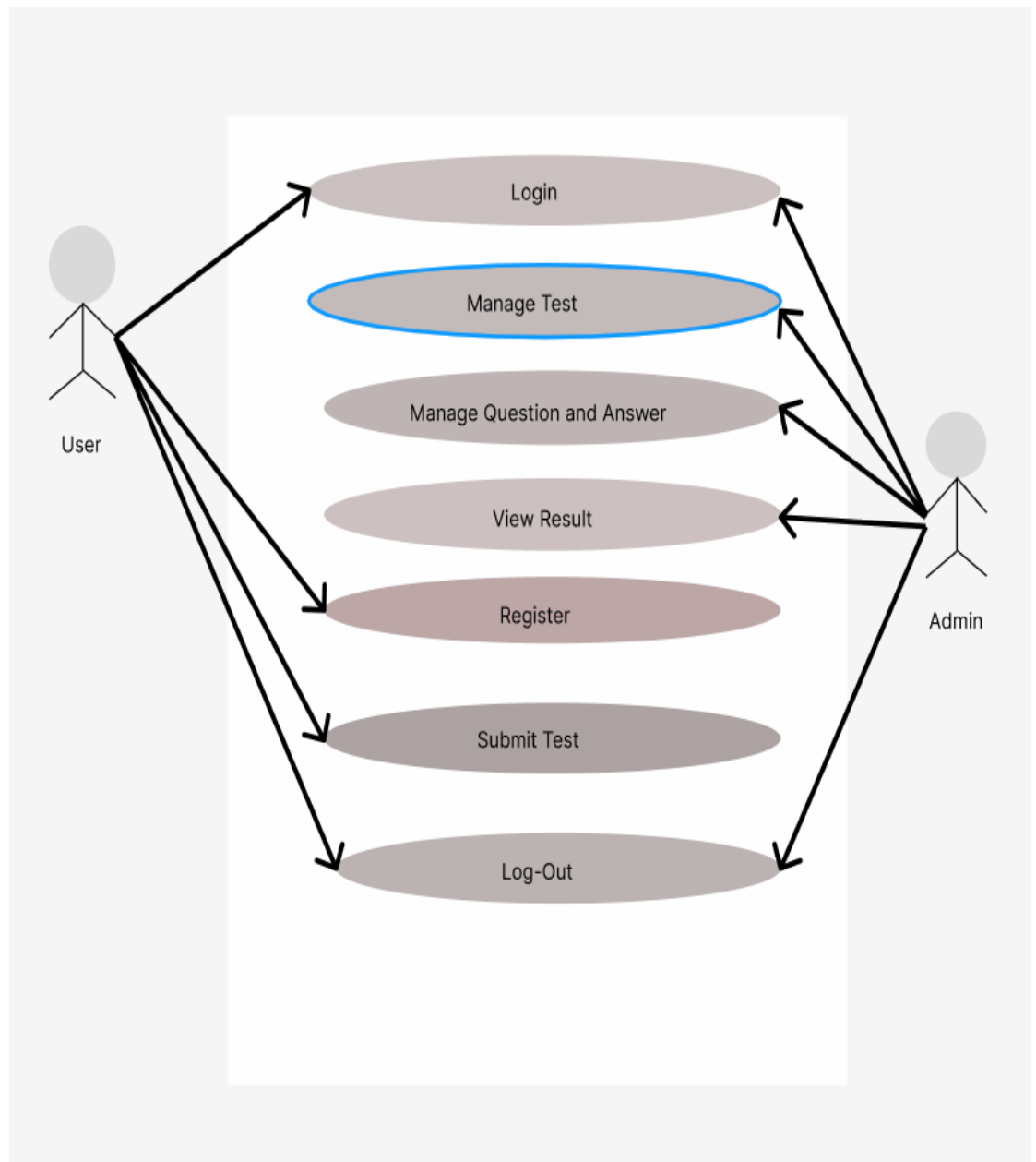


Figure 3.1.3.1

### **3.1.4 Programming**

#### **HTML**

The hypertext mark-up language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. It is often assisted by technologies such as CSS and scripting languages such as JavaScript.

#### **Advantages**

It is easy to learn.

Every browser supports HTML Language.

HTML is light weighted and fast to load.

#### **CSS**

Cascading style sheet is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML. CSS is a cornerstone technology of the World Wide Web.

#### **Advantages**

Improve the browsing speed.

It can be used on various devices.

Wider variety of design options.

#### **JavaScript**

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

#### **Advantages**

Fast speed

Easy to learn

Versatility

Popularity

## **Bootstrap**

Bootstrap is a free and open source CSS framework directed at responsive, front end web development. It contains HTML, CSS and java script based design templates for forms, buttons, navigation, etc.

### **Advantages**

- Open source
- Easy to use
- Save lots of time
- Compatible with browsers

## **PHP**

PHP hypertext pre-processor is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML.PHP pages contains HTML with embedded code .

### **Advantages**

- It's open-source and free from cost.
- It is platform-independent.
- It helps in managing code easily.

### **3.1.5 Analysis**

According to my survey, some institutes administer they test manually because there is no online software available. As a result, they spend more time manually preparing the test materials.

The goal of this project is to build an “Online MCQ Exam”. Through this project, staff members and administrations and can conduct an online exam, take the test, and view the result list. User must first register before they can login, answer exam question, and receive the result right away. We used Full Stack Languages, such as HTML, CSS, JavaScript, PHP, MySQL, and others, will be used to construct this project.

## 3.1.5 Process Design

### List of tables

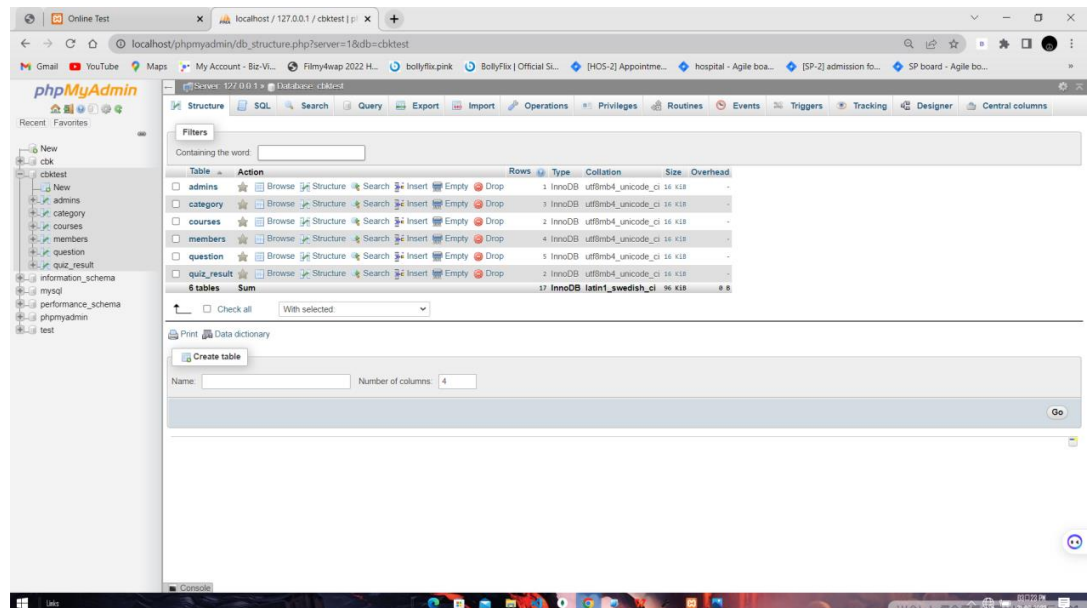


Figure 3.1.6.1

### Admin table

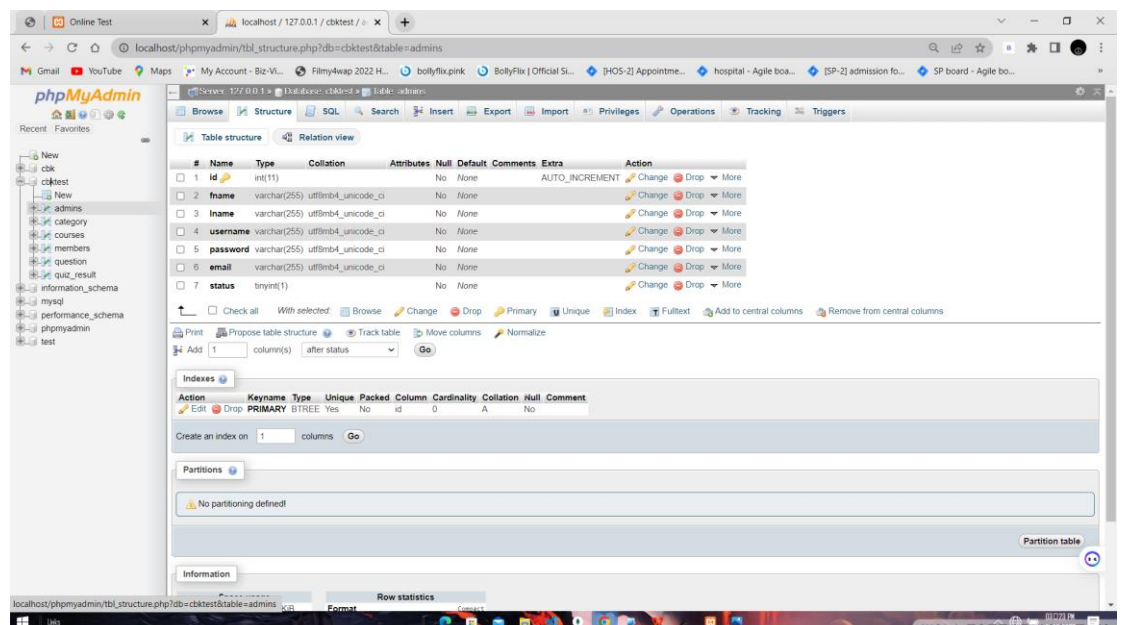


Figure 3.1.6.2

## Category table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	catid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	catname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	totalque	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	course_id	int(11)			No	None			Change Drop More
5	course_year_id	int(11)			No	None			Change Drop More
6	semester_id	int(11)			No	None			Change Drop More

Figure 3.1.6.3

## Courses table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	course_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	status	tinyint(4)			No	0			Change Drop More

Figure 3.1.6.4

## Members table

The screenshot shows the phpMyAdmin interface for the 'members' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	fname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	lname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	username	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
7	phone	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
8	address	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
9	course_id	int(11)			No	None			Change Drop More
10	course_year_id	int(11)			No	None			Change Drop More
11	semester_id	int(11)			No	None			Change Drop More
12	status	tinyint(1)			No	0			Change Drop More

The 'id' column is the primary key. The 'status' column is a tinyint(1) with a default value of 0.

Figure 3.1.6.5

## Questions table

The screenshot shows the phpMyAdmin interface for the 'question' table. The table structure is as follows:

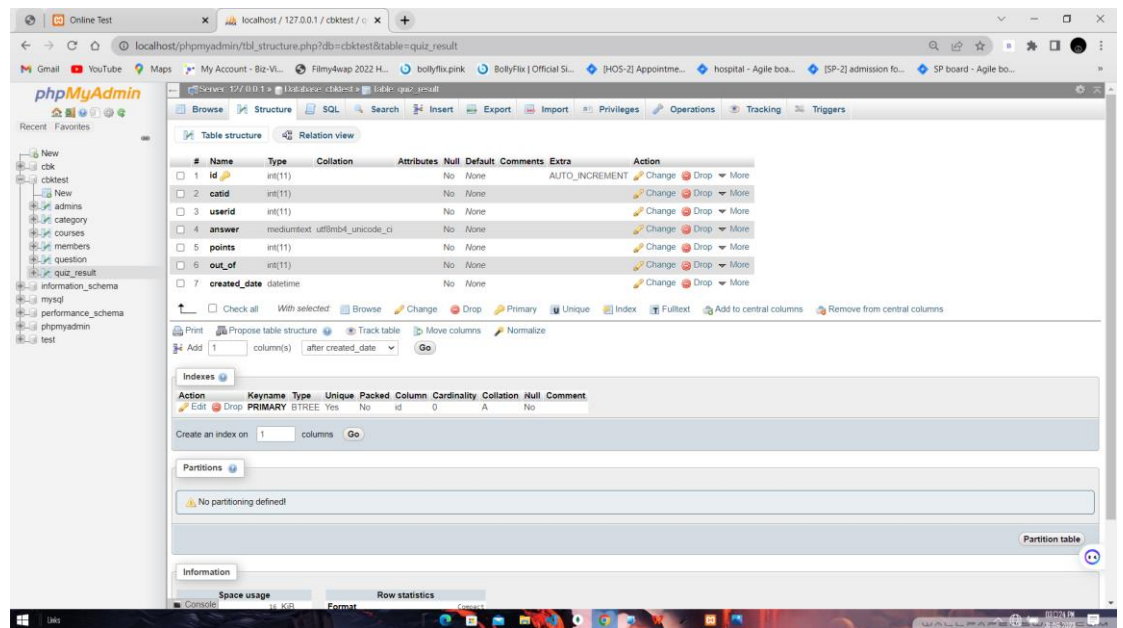
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	quid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	catid	int(11)			No	None			Change Drop More
3	testimage	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	question	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	opt1	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	opt2	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
7	opt3	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
8	opt4	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
9	trueans	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More

The 'quid' column is the primary key. The 'trueans' column is a varchar(255) column.

Figure 3.1.6.6



## Quiz result table



The screenshot shows the phpMyAdmin interface for a database named 'cbktest'. The 'quiz\_result' table is selected, and its structure is displayed in the 'Table structure' tab. The table has 7 columns: id, catid, userid, answer, points, out\_of, and created\_date. The 'id' column is the primary key. Below the table structure, the 'Indexes' tab is active, showing a primary index on the 'id' column. The 'Partitions' tab shows no partitioning defined. The 'Information' tab shows space usage and row statistics.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	catid	int(11)			No	None			Change Drop More
3	userid	int(11)			No	None			Change Drop More
4	answer	mediumtext	utf8mb4_unicode_ci		No	None			Change Drop More
5	points	int(11)			No	None			Change Drop More
6	out_of	int(11)			No	None			Change Drop More
7	created_date	datetime			No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Create an index on 1 columns Go

Partitions: No partitioning defined

Information: Space usage, Row statistics

Figure 3.1.6.7

## 3.1.6 Product Design

### Index page

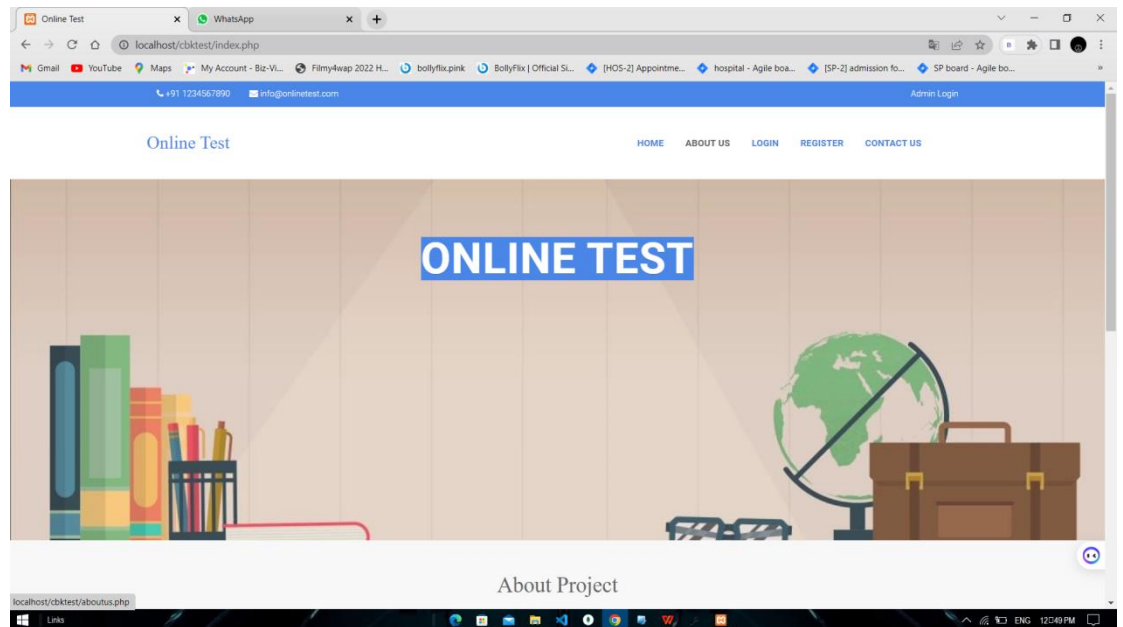


Figure 3.1.7.1

### Dashboard

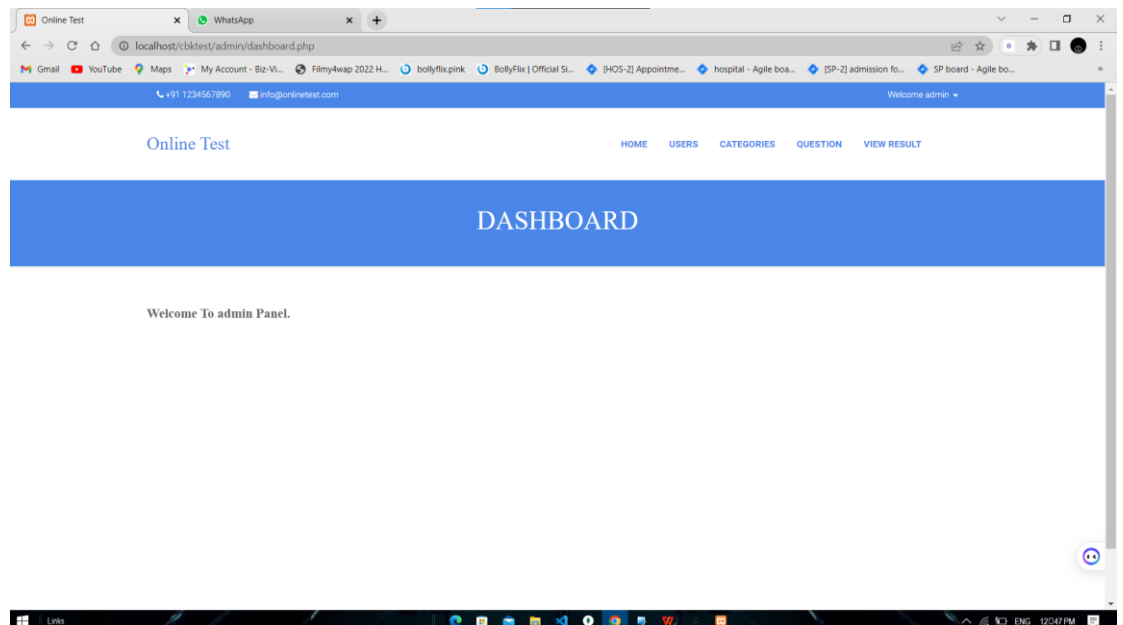


Figure 3.1.7.2

## Add category page

Online Test

HOME USERS CATEGORIES QUESTION VIEW RESULT

### ADD CATEGORY

Enter Category Name

Enter Total Questions

Select Course

Select Course Year

Select Semester

ADD CATEGORY

Figure 3.1.7.3

## View category page

Online Test

HOME USERS CATEGORIES QUESTION VIEW RESULT

### VIEW CATEGORY

Category Name	Total Question	Course	Course Year	Semester	Actions
python	2	Computer Science & Engineering	Third Year	VI	<a href="#">Edit</a>
java	4	Computer Science & Engineering	First Year	II	<a href="#">Edit</a>

© 2023 Online Test. All Rights Reserved.

Figure 3.1.7.4

## Add question page

Online Test

HOME USERS CATEGORIES QUESTION VIEW RESULT

### ADD QUESTION

Select Category

Choose file | No file chosen

Enter Question

Enter Option 1

Enter Option 2

Enter Option 3







Figure 3.1.7.5

## View questions page

Online Test

HOME USERS CATEGORIES QUESTION VIEW RESULT

### VIEW QUESTION

Category Name	Photo	Question	Opt 1	Opt 2	Opt 3	Opt 4	Answer	Actions
java		Array in java are	int	Objects	64 and 64	char[] ch= new char[]	int	 
python		Number of primitive data types in java	Object references	A copy of the array	Primitive data type	Copy of first element	A copy of the array	 
python		Select the valide statement.	char[] ch= new char(5)	32 and 32	Length of the array	64 and 32	Length of the array	 

© 2023 Online Test. All Rights Reserved.

Figure 3.1.7.6

## View result page

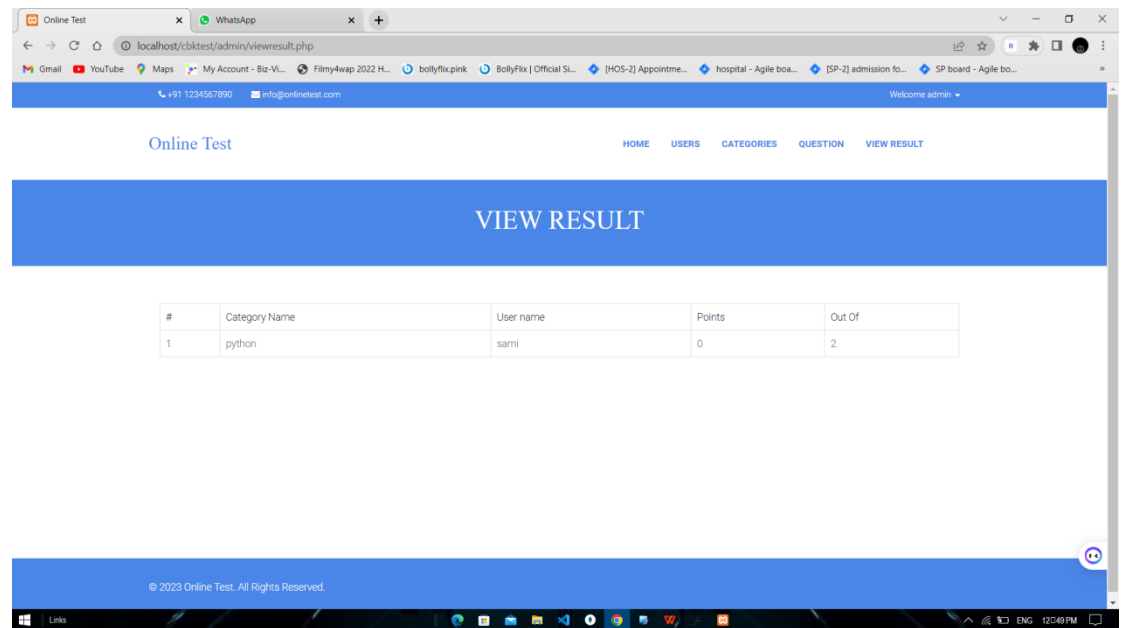


Figure 3.1.7.7

## Registration page

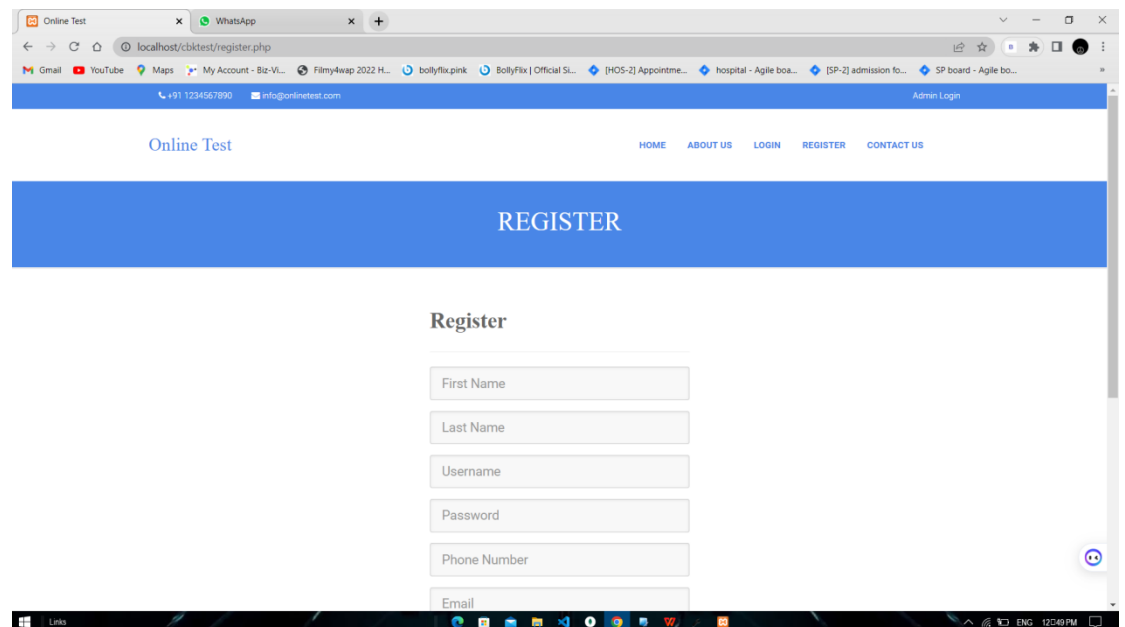
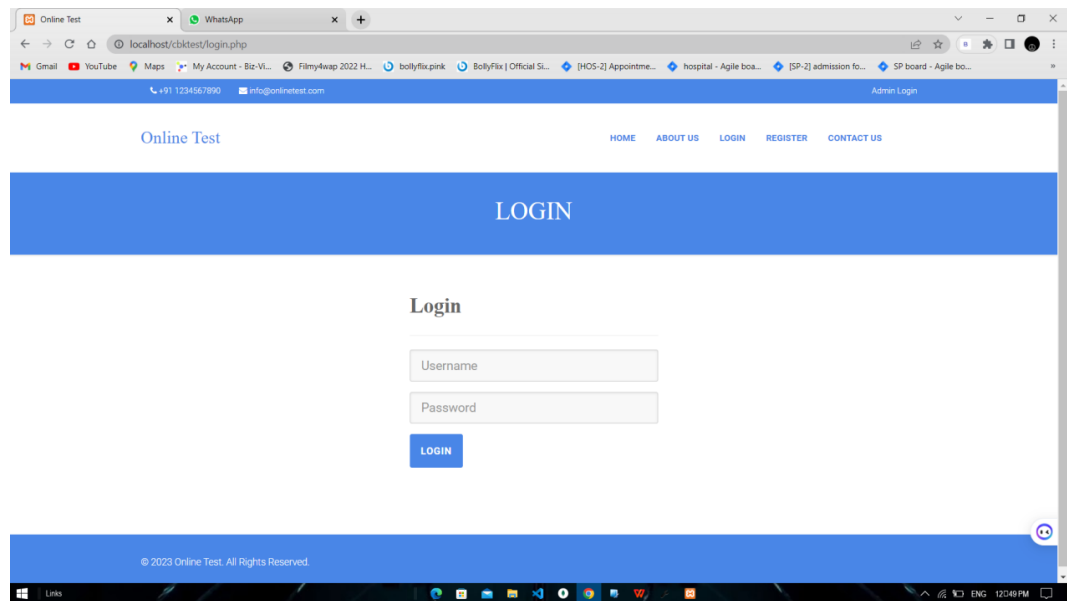


Figure 3.1.7.8

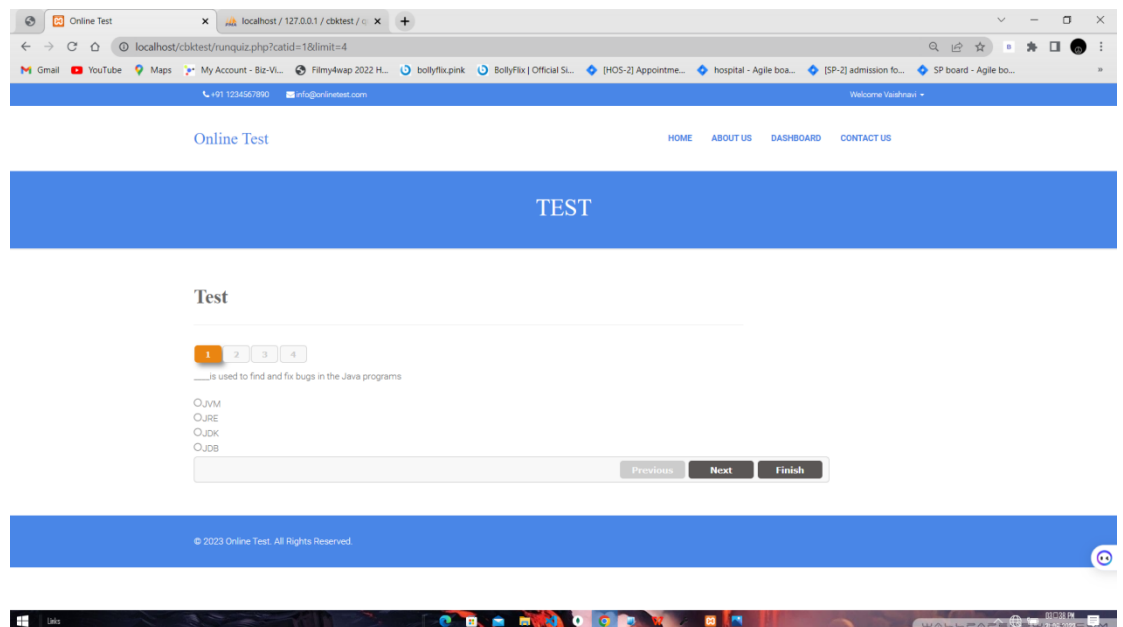
## User login page



The screenshot shows a web browser window with the URL `localhost/cbctest/login.php`. The page has a blue header with the text "Online Test" and navigation links: HOME, ABOUT US, LOGIN, REGISTER, CONTACT US. Below the header is a large blue banner with the word "LOGIN" in white. Underneath the banner is a "Login" section with two input fields: "Username" and "Password". A blue "LOGIN" button is positioned below the password field. At the bottom of the page, there is a blue footer with the text "© 2023 Online Test. All Rights Reserved." and a small circular icon on the right.

Figure 3.1.7.9

## Test page



The screenshot shows a web browser window with the URL `localhost/cbctest/runquiz.php?catid=1&limit=4`. The page has a blue header with the text "Online Test" and navigation links: HOME, ABOUT US, DASHBOARD, CONTACT US. Below the header is a large blue banner with the word "TEST" in white. Underneath the banner is a "Test" section. It features a progress bar with four numbered buttons (1, 2, 3, 4), where button 1 is highlighted. Below the progress bar is a text input field containing the question: "\_\_\_ is used to find and fix bugs in the Java programs". Below the input field are four radio button options: O JVM, O JRE, O JDK, and O JDB. At the bottom of the test section are three buttons: "Previous", "Next", and "Finish". At the bottom of the page, there is a blue footer with the text "© 2023 Online Test. All Rights Reserved." and a small circular icon on the right.

Figure 3.1.7.10

### **3.1.8 Fabrication**

#### **Admin/Staff login**

In admin login we have collected the requirements and analysed the requirements. Designed the form as per user requirements. We have validated the form. First we have created database then we gave connection to the database. At last we have tested the validation and database connection of the form.

#### **Manage Test**

In manage test we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the page. Then we created the table to manage test as per user and gave the connection to the database. At last we tested the validation and database of the page.

#### **Manage question and answer**

In manage question and answers we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the page. Then we created the table to manage question and answer as per user and gave the connection to the database. At last we tested the validation and database of the page.

#### **View result**

In view result we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the page. Then we created the table to result as per user and gave the connection to the database. At last we tested the validation and database connection of the page to view the result.

#### **User/Student registration**

In user/student registration we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the form. Then we created the table to user/student registration as per user and gave the connection to the database. At last we tested the validation and database of the form.

**User login**

In user login we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the form. Then we created the table to user login as per user and gave the connection to the database. At last we tested the validation and database of the form.

**Submit test**

In submit test we collected the requirements from user and analysed the requirements. Designed the page as per user requirements. We have validated the page. Then we created the table to submit test as per user and gave the connection to the database. At last we tested the validation and database of the page.



## CHAPTER 4

### 4.1 Test and Validation

#### 4.1.1 Test Plan

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

#### 4.1.2 Software testing can be divided into two steps:

**Verification:** it refers to the set of tasks that ensure that the software correctly implements a specific function.

**Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

#### 4.1.3 Testing Types

##### **Manual Testing: -**

Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

##### **Unit Testing: -**

Unit testing is a software development process in which the smallest testable parts of an application, are called units. The main objective of unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process. If done correctly, unit tests can detect early flaws in code which may be more difficult to find in later testing stages.

##### **Integration Testing: -**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

**White Box Testing: -**

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

**Black Box Testing: -**

The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

## 4.1.2 Test Approach

Table 4.1.2.1

### Admin Login

Test Case id	Description	Test Step	Test data	Expected Result	Actual Result	Status
TC01	Check admin login with valid data	Enter Username Enter Password Click login	Username=admin Password=admin	Admin should login to the form	As expected	Pass

Table 4.1.2.2

### Manage Test

Test Case id	Description	Test step	Test data	Expected Result	Actual Result	Status
TC02	Add the test in that category name should be text and total numbers of questions should be numbers only.	Enter category Enter total number of questions Select course, course year and semester	Category=java Total number of questions =10 Course = computer science Course year=2 Semester =4	Admin should add the category to the form successfully.	As expected	Pass

Table 4.1.2.3

### Manage question and answer

Test Case id	Description	Test Step	Test data	Expected Result	Actual Result	Status
TC03	Add questions in that question field, options and correct answer should not be empty	Select category Enter question Enter options Select correct answer	Category=java Question=data types in java Options= 1)6 2)8 3)9 4)5 Correct answer=8	Admin should add questions to the test successfully.	As expected	Pass

Table 4.1.2.4

**User/Student Registration**

Test Case id	Description	Test Step	Test data	Expected Result	Actual Result	Status
TC04	Check user registration with valid data	Enter first name Enter last name Enter username Enter password Enter phone Enter email Select course Select course year Select semester	Fname= vaishnavi Lname=Tharakar Username=Vaishnavi Password=***** Phone=9986609349 <a href="mailto:vaish@gmail.com">Email=vaish@gmail.com</a> Course=CSE Course year=2 Semester=4	User should be able to register to the form successfully.	As expected	Pass

Table 4.1.2.5

**User/Student login**

Test Case id	Description	Test Step	Test data	Expected Result	Actual Result	Status
TC05	Check user login with valid data	Enter Username Enter Password Click login	Username=Vaishnavi Password=*****	User should login to the form	As expected	Pass

Table 4.1.2.6

**Submit test**

Test Case id	Description	Test Step	Test data	Expected Result	Actual Result	Status
TC06	Validate user entered answers	Select answers	Correct answer=8	User should be able to get result	As expected	Pass

### **4.1.3 Features Tested**

Admin Login

Manage test

Manage question and answer

View result

User/Student registration

User/Student login

Submit test

### **4.1.4 Features not Tested**

We have not tested the project on cloud server.

### **4.1.5 Findings**

#### **Admin login**

If admin doesn't provide any username and password in username and password field, we will receive a pop-up message as please enter username and password, and we expected the same result, hence the test is passed.

#### **Manage test**

If admin provide numbers and special character in the category name field and in total numbers of questions, they will receive a pop-up message as category name should not be numeric and total number questions should be numeric only, and we expected the same result, hence the test is passed.

#### **Manage question and answer**

If admin doesn't provide category name in category field and question in question field, we will receive a pop-up message as please enter category and please enter question, and we expected the same result, hence the test is passed.

**User/student registration**

If user doesn't provide any username and password in username and password field, we will receive a pop-up message as please enter username, please enter password and we expected the same result, hence the test is passed.

**User login**

If user doesn't provide any username and password in username and password field, we will receive a pop-up message as please enter username and password, and we expected the same result, hence the test is passed

**Submit test**

If user doesn't click submit button he will not able to get result, here he have to click submit button as we expected the same result, hence the test is passed.

## **4.1.6 Inference**

### **Admin Login**

Admin login is very important in our project because admin manages everything related and users. User also plays an important role in Online MCQ Exam. User is dependent on admin to attend the exam.

### **Manage test**

Manage test is completely dependent on admin. If admin didn't login he is not able to add or delete the tests to the form. Here it is totally dependent on admin login.

### **Manage question and answer**

Manage question and answer is totally dependent on manage test. If admin didn't add test to the form then he is not able to add questions and answers to the test. So it is important to add test to the form then only admin able to add questions and answers.

### **View Result**

View result is dependent on manage test and manage question and answers. If admin didn't add tests and questions to the form then only he is able to see the result. Without adding questions and tests he is not able to see the result.

### **User/Student Registration**

User registration is dependent on Admin login. Here user must register with his details with correct details. Here it is dependent on admin login. Admin must allow user to login then only user is able to login. If admin didn't allow user then he is not able to login to the test.

### **User/Student Login**

User login is dependent on user registration. If user didn't registered with his correct details then his not able to login. If you not entered correct details admin will not allow you to login. So you have to enter correct detail while registration.

### **Submit Test**

Submit test is dependent on user login. If user didn't login to the form he is not able to attend the test. So here user first login using username and password which he has entered while registration then only he is able to submit the test. After submitting the test he will get the result.



## **CHAPTER 5**

### **5.1 Business aspects**

#### **5.1.1 The market and economic outlook of the project**

Online MCQ Exam is a technology-driven way to simplify examination activities like defining exam patterns with question banks, defining exam timers, objective/ subjective question sections, and conducting exams using a computer or mobile devices in a paperless manner.

Online MCQ Exam is a cost-effective, saleable way to convert traditional pen and paper-based exams to online and paperless mode. Candidates can appear for the exam using any desktop, laptop, or mobile device with a browser. Exam results can be generated instantly for the objective type of questions

This is developed using HTML, CSS, Bootstrap, JavaScript, PHP and MySQL database. Some changes could be done here to make it more reliable, more automatic and providing more features.

The goal of this project is to build an "online MCQ exam." Through this project, staff members and administrators can conduct an online exam, take the test, and view the result list. Users must first register before they can log in, answer exam questions, and receive their results right away. We have used Full stack languages, such as HTML, CSS, JavaScript, PHP, MySQL, and others, will be used to construct this project.

#### **5.1.2 Features**

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Student can submit online test and get result easily.

### **5.1.3 How does the product/service fit into the competitive landscape?**

Online MCQ exams can provide a quick and efficient way for teachers to assess student learning and understanding. It can also help students to practice and improve their test-taking skills, as well as receive immediate feedback on their performance. Additionally, online MCQ exams can save time and resources for both teachers and students. Online MCQ exams can be customized to suit the needs of individual students in several ways. One way is to tailor the difficulty level of the questions based on the student's performance in previous assessments. Another way is to allow students to choose the topics they want to be tested on, so they can focus on areas where they need more practice. Additionally, online MCQ exams can be timed or untimed, depending on the student's preference and ability. By customizing online MCQ exams, teachers can provide a more personalized learning experience for their students, which can lead to better engagement and improved academic performance.

### **5.1.4 Possible capstone project clients /customers**

**Educational Institutions:** Schools, colleges, universities, and other educational institutions often require online MCQ tests for their students. These tests may be used for assessments, examinations, quizzes, or entrance exams.

**E-Learning Platforms:** Online learning platforms that offer courses and training programs may utilize MCQ tests to assess learners' knowledge and progress. These platforms could include Moo Cs (Massive Open Online Courses), corporate training portals, or specialized e-learning platforms.

**Recruitment and Hiring Agencies:** Companies and recruitment agencies often utilize online MCQ tests as part of their hiring process. These tests help assess candidates' skills and knowledge related to specific job roles.

**Certification and Licensing Bodies:** Organizations responsible for issuing certifications and licenses may require online MCQ tests to evaluate candidates' eligibility and competence. Examples include professional certification bodies, regulatory agencies, and industry associations.

**Government Agencies:** Government bodies responsible for conducting examinations, such as civil service exams or entrance tests for educational institutions, may opt for online MCQ test projects to streamline the assessment process and enhance efficiency.

## 5.2 Financial Consideration

### 5.2.1 Capstone project budget: 43,700/-

### 5.2.2 Cost capstone projections needed for either for profit/non-profit options.

Table 5.2.2.1

<b>Estimations</b>	<b>Cost(Rupees)</b>
Labor Cost	25,200
Cost of material	8,500
Net Profit	10,000
<b>Total</b>	43,700

## 5.3 Conclusion and Recommendations

### 5.3.1 Conclusion

Admin/staff login, Manage test, Manage question and answer, View result, User/student registration, User login, Submit test, and error-free and user-friendly are all modules we will implement for Online MCQ Test in accordance with the summary.

### 5.3.2 Future work

We can live the project to any institute.

We can develop the mobile application.

## Program code

### Index.php

```
<!DOCTYPE HTML>
<html class="no-js">
<head>
<!-- Basic Page Needs
===== -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Online Test</title>
<!-- Mobile Specific Metas
===== -->
<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0">
<meta name="format-detection" content="telephone=no"/>
<!-- CSS
===== -->
<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
<link href="css/style.css" rel="stylesheet" type="text/css"/>
<link      href="plugins/prettyphoto/css/prettyPhoto.css"      rel="stylesheet"
type="text/css"/>
```

```

<link      href="plugins/owl-carousel/css/owl.carousel.css"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/owl-carousel/css/owl.theme.css"        rel="stylesheet"
type="text/css"/>
<link rel="stylesheet" type="text/css" href="plugins/rs-plugin/css/settings.css"
media="screen" />
<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->
<!-- Color Style -->
<link class="alt" href="colors/purple.css" rel="stylesheet" type="text/css">
<!-- SCRIPTS
===== -->
<script src="js/modernizr.js"></script><!-- Modernizr -->
<?php
require_once "testhelper.php";
$helper = new TestHelper();
?>
</head>
<body>
<!-- Start Body Container -->
<div class="body footer-style4">
    <?php
    require_once "header.php";
    ?>
    <!-- End Header -->
    <!-- Start Content -->
    <div class="main" role="main">
        <div id="content" class="content full">
            <div class="rev-slider-container">
                <div class="tp-banner-container">
                    <div class="tp-banner" >
                        <ul>

                            <!-- SLIDE -->

```

```

        <li data-delay="4000" data-masterspeed="600" data-slotamount="7" data-
transition="fade">
            <!-- MAIN IMAGE -->
            
            <!-- LAYER NR. 1 -->
            <div class="tp-caption centered_white_huge with_colored_background
lft tp-resizeme" data-endspeed="300" data-easing="Power4.easeOut" data-
start="400" data-speed="600" data-y="80" data-hoffset="0" data-
x="center"><strong>Online test</strong></div>
        </li>
        <li data-delay="4000" data-masterspeed="600" data-slotamount="7" data-
transition="fade">
            <!-- MAIN IMAGE -->
            
            <!-- LAYER NR. 1 -->
            <div class="tp-caption centered_white_huge with_colored_background
lft tp-resizeme" data-endspeed="300" data-easing="Power4.easeOut" data-
start="400" data-speed="600" data-y="80" data-hoffset="0" data-
x="center"><strong>Online test</strong></div>
        </li>
    </ul>
</div>
</div>
<!--/Portfolio Plus Filters -->

<footer class="heading-fs-bg lgray-color no-shadow margin-50">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <h2 class="no-strong margin-0">About Project</h2>
            </div>
        </div>
    </div>

```

```
</div>
</footer>
```

```
<div class="container margin-50">
  <div class="row">
    <div class="col-md-12 col-sm-12">
      <p>The goal of this project is to build an "online MCQ exam." Through this
project, staff members and administrators can conduct an online exam, answer
exam questions, and view the result list. Users must first register before they can
log in, take the exam, and receive their results right away. Full stack languages,
such as HTML, CSS, JavaScript, PHP, MySQL, and others, will be used to
construct this project.</p>
    </div>
  </div>
</div>
```

```
<hr class="fw">
<div class="container margin-30">
  <div class="row">
    <div class="col-md-12">
      <header class="heading-with-icon">
        <h2>Features</h2>
        <hr/>
      </header>
    </div>
    <section class="features-ico-left alt">
      <div class="col-md-3 col-sm-6">
        <div class="block margin-30"> <i class="fa ic-sm accent-color color-text
fa-lg fa-star"></i>
          <h5 class="short">Admin Login</h5>
          <p>Here admin can login to the form using admin username and
password. After logging in he can add the tests to conduct online exam, he can add
```



or delete questions and answers to the particular test. He can allow user who wants to login to the form and also he can manage the result. </p>

</div>

</div>

<div class="col-md-3 col-sm-6">

<div class="block margin-30"> <i class="fa ic-sm accent-color color-text fa-lg fa-star"></i>

<h5 class="short">User Registration</h5>

<p>User can register himself by adding his information. User have to add detailed information with username and password, so he can easily login to the form. </p>

</div>

</div>

<div class="col-md-3 col-sm-6">

<div class="block margin-30"> <i class="fa ic-sm accent-color color-text fa-lg fa-star"></i>

<h5 class="short">User Login</h5>

<p>After registration user can login to the form using username and password which he had entered while registration. After logging in he is able to attend the test.</p>

</div>

</div>

<div class="col-md-3 col-sm-6">

<div class="block margin-30"> <i class="fa ic-sm accent-color color-text fa-lg fa-star"></i>

<h5 class="short">Submit Test</h5>

<p>After login user can enter into the test, and he can attend the test. After clearing all questions he can submit the test and get result quickly without any issues.</p>

</div>

</div>

</section>

</div>

```

        </div>
    </div>
</div>
<!-- End Footer -->
<?php
require_once "footer.php";
?>
<!-- End footer -->
<a id="back-to-top"><i class="fa fa-angle-double-up"></i></a> </div>
<!-- End Body Container -->
<script src="js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->
<script src="plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="plugins/owl-carousel/js/owl.carousel.min.js"></script>
<script src="plugins/page-scroller/jquery.pagescroller.js"></script>
<script src="js/helper-plugins.js"></script> <!-- Plugins -->
<script src="js/bootstrap.js"></script> <!-- UI -->
<script src="js/init.js"></script> <!-- All Scripts -->
<script src="plugins/rs-plugin/js/jquery.themepunch.plugins.min.js"></script>
<script src="plugins/rs-plugin/js/jquery.themepunch.revolution.min.js"></script>
<script src="js/revolution-slider-init.js"></script> <!-- Revolutions Slider
Intialization -->
<!-- End Js -->

</body>
</html>

```

## Admin login.php

```
<!DOCTYPE HTML>
<html class="no-js">
<head>
<!-- Basic Page Needs
===== -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Online Test</title>
<!-- Mobile Specific Metas
===== -->
<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0">
<meta name="format-detection" content="telephone=no"/>
<!-- CSS
===== -->
<link href="../css/bootstrap.css" rel="stylesheet" type="text/css"/>
<link href="../css/style.css" rel="stylesheet" type="text/css"/>
<link      href="../plugins/prettyphoto/css/prettyPhoto.css"      rel="stylesheet"
type="text/css"/>
<link      href="../plugins/owl-carousel/css/owl.carousel.css"      rel="stylesheet"
type="text/css"/>
<link      href="../plugins/owl-carousel/css/owl.theme.css"      rel="stylesheet"
type="text/css"/>
<link rel="stylesheet" type="text/css" href="../plugins/rs-plugin/css/settings.css"
media="screen" />
<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->
<!-- Color Style -->
<link class="alt" href="../colors/purple.css" rel="stylesheet" type="text/css">
<!-- SCRIPTS
===== -->
<script src="js/modernizr.js"></script><!-- Modernizr -->
```

```

<script type="text/javascript">
function validate_form()
{
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;

    if(username=="")
    {
        alert("Please Enter User Name.");
        return false;

    }
    else if(password=="")
    {
        alert("Please Enter Password.");
        return false;

    }
}
</script>

```

```

<?php
require_once "adminhelper.php";
$helper = new AdminHelper();
?>

```

```

</head>
<body>
<!-- Start Body Container -->
<div class="body">
    <!-- Start Header -->
    <?php
        require_once "header.php";
    ?>

```

```

<!-- End Header -->

<!-- Start Content -->
<div class="main" role="main">
  <div id="content" class="content page-content full">
    <header class="page-header flexible parallax text-align-center parallax-
overlay">
      <section>
        <div class="container">
          <div class="row">
            <div class="col-md-12">
              <h1>Login</h1>
            </div>
          </div>
        </div>
      </section>
    </header>
    <div class="container">
      <div class="row">
        <div class="col-md-4 col-md-offset-4">
          <h2><strong>Login</strong></h2>
          <hr/>
          <?php
            if($_GET['error'])
            {
              ?>
              <div class="alert alert-error fade in">
                <a href="#" data-dismiss="alert" class="close">&times;</a>
                <?php echo "Invalid Details";?>
              </div>
            }
          <?php
            ?>

```

```

        <form name="adminlogin" id="adminlogin" action="checklogin.php"
method="post" onsubmit="return validate_form();" >
        <div class="row">
            <div class="form-group">
                <div class="col-md-12">
                    <input type="text" placeholder="Username" class="form-control
input-lg" name="username" id="username">
                </div>
            </div>
        </div>
        <div class="row">
            <div class="form-group">
                <div class="col-md-12">
                    <input type="password" placeholder="Password" class="form-control
input-lg" name="password" id="password">
                </div>
            </div>
        </div>
        <div class="row">
            <div class="form-group">
                <div class="col-md-12">
                    <input type="submit" value="Login" class="btn btn-primary btn-lg"
name="submit">
                </div>
            </div>
        </div>
    </form>
</div>
<!-- Start Sidebar -->
<aside class="col-md-3 sidebar right-sidebar">

</aside>

```

```

        </div>
    </div>
</div>
</div>

<?php
    require_once "footer.php";
?>
</div>
<!-- End Body Container -->
<script src="../../js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->
<script src="../../plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="../../plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="../../plugins/owl-carousel/js/owl.carousel.min.js"></script>
<script src="../../plugins/page-scroller/jquery.pagescroller.js"></script>
<script src="../../js/helper-plugins.js"></script> <!-- Plugins -->
<script src="../../js/bootstrap.js"></script> <!-- UI -->
<script src="../../js/init.js"></script> <!-- All Scripts -->
<!-- End Js -->
</body>
</html>

```

## User registration page

```
<!DOCTYPE HTML>
<html class="no-js">
<head>
<!-- Basic Page Needs
===== -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Online Test</title>
<!-- Mobile Specific Metas
===== -->
<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0">
<meta name="format-detection" content="telephone=no"/>
<!-- CSS
===== -->
<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
<link href="css/style.css" rel="stylesheet" type="text/css"/>
<link      href="plugins/prettyphoto/css/prettyPhoto.css"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/owl-carousel/css/owl.carousel.css"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/owl-carousel/css/owl.theme.css"      rel="stylesheet"
type="text/css"/>
<link rel="stylesheet" type="text/css" href="plugins/rs-plugin/css/settings.css"
media="screen" />
<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->
<!-- Color Style -->
<link class="alt" href="colors/purple.css" rel="stylesheet" type="text/css">
<!-- SCRIPTS
===== -->
<script src="js/modernizr.js"></script><!-- Modernizr -->
```



```

<script type="text/javascript">
function validate_form()
{

    var fname  = document.getElementById("fname").value;
    var lname  = document.getElementById("lname").value;
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    var phone   = document.getElementById("phone").value;
    var email   = document.getElementById("email").value;


    var course_id    = document.getElementById("course_id").value;
    var course_year_id = document.getElementById("course_year_id").value;
    var semester_id   = document.getElementById("semester_id").value;


    var validchar = /^[A-Za-z]+$;/;
    if(fname=="")
    {
        alert("Please Enter First Name.");
        return false;
    }
    else if(!validchar.test(fname))
    {
        alert("First Name should not be numeric.");
        return false;
    }
    else if(lname=="")
    {
        alert("Please Enter Last Name.");
        return false;
    }
    else if(!validchar.test(lname))
    {
        alert("Last Name should not be numeric.");
    }
}

```

```

        return false;
    }
    else if(username=="")
    {
        alert("Please Enter User Name.");
        return false;
    }
    else if(password=="")
    {
        alert("Please Enter Password.");
        return false;
    }
    else if(phone=="")
    {
        alert("Please Enter Phone Number.");
        return false;
    }
    else if(isNaN(phone))
    {
        alert("Phone Number should be numeric.");
        return false;
    }
    else if(checkInternationalPhone(phone)==false)
    {
        alert("Please Enter a Valid Phone Number");
        return false;
    }
    else if(email=="")
    {
        alert("Please Enter Email Address.");
        return false;
    }
    else if(validateEmail(email))
    {

```

```

        alert("Please Enter Valid Email Address.");
        return false;
    }
    else if(course_id=="")
    {
        alert("Please Select Course.");
        return false;
    }
    else if(course_year_id=="")
    {
        alert("Please Select Course Year.");
        return false;
    }
    else if(semester_id=="")
    {
        alert("Please Select Semester.");
        return false;
    }
}

function validateEmail(email)
{
    var atpos = email.indexOf("@"); // The indexOf() method returns the position
of the first occurrence of a specified value in a string. // Default value of start is 0
    //alert(atpos);

    var dotpos = email.lastIndexOf("."); // The lastIndexOf() method returns the
position of the last occurrence of a specified value in a string. // Default value of
start is 0
    //alert(dotpos);

    if((atpos<1) || (dotpos<(atpos+2)) || (dotpos+2>=email.length))
    {
        return true;
    }
}

```

```

else
{
    return false;
}
}

// Declaring required variables
var digits = "0123456789";
// non-digit characters which are allowed in phone numbers
var phoneNumberDelimiters = "() - ";
// characters which are allowed in international phone numbers
// (a leading + is OK)
var validWorldPhoneChars = phoneNumberDelimiters + "+";
// Minimum no of digits in an international phone no.
var minDigitsInIPhoneNumber = 10;

function isInteger(s)
{
    var i;
    for (i = 0; i < s.length; i++)
    {
        // Check that current character is number.
        var c = s.charAt(i);
        if (((c < "0") || (c > "9"))) return false;
    }
    // All characters are numbers.
    return true;
}

function trim(s)
{
    var i;
    var returnString = "";
    // Search through string's characters one by one.
    // If character is not a whitespace, append to returnString.
    for (i = 0; i < s.length; i++)

```

```

{
    // Check that current character isn't whitespace.
    var c = s.charAt(i);
    if (c != " ") returnString += c;
}
return returnString;
}

```

```

function stripCharsInBag(s, bag)
{
    var i;
    var returnString = "";
    // Search through string's characters one by one.
    // If character is not in bag, append to returnString.
    for (i = 0; i < s.length; i++)
    {
        // Check that current character isn't whitespace.
        var c = s.charAt(i);
        if (bag.indexOf(c) == -1) returnString += c;
    }
    return returnString;
}

```

```

function checkInternationalPhone(strPhone){
    var bracket=3;
    strPhone=trim(strPhone);
    if(strPhone.indexOf("+")>1) return false;
    if(strPhone.indexOf("-")!=-1)bracket=bracket+1;
    if(strPhone.indexOf("(")!=-1 && strPhone.indexOf("(")>bracket)return false;
    var brchr=strPhone.indexOf("(");
    if(strPhone.indexOf("(")!=-1 && strPhone.charAt(brchr+2)!=")")return false;
    if(strPhone.indexOf("(")==-1 && strPhone.indexOf(")")!=-1)return false;
    s=stripCharsInBag(strPhone,validWorldPhoneChars);
    return (isInteger(s) && s.length >= minDigitsInIPhoneNumber);
}

```

```
</script>
```

```
<?php
```

```
    require_once "testhelper.php";
```

```
    $helper = new TestHelper();
```

```
    $msg = "";
```

```
    $class = "";
```

```
    if($_POST)
```

```
    {
```

```
        $result = $helper->saveRegister();
```

```
        if($result)
```

```
        {
```

```
            $msg = "Registration successfully.";
```

```
            $class = 'success';
```

```
        }
```

```
    else
```

```
    {
```

```
        $msg = "Sorry Please try again...";
```

```
        $class = 'alert';
```

```
    }
```

```
    }
```

```
?>
```

```
</head>
```

```
<body>
```

```
<!-- Start Body Container -->
```

```
<div class="body">
```

```

<!-- Start Header -->
<?php
    require_once "header.php";
?>
<!-- End Header -->
<!-- Start Content -->
<div class="main" role="main">
    <div id="content" class="content page-content full">
        <header class="page-header flexible parallax text-align-center parallax-
overlay">
            <section>
                <div class="container">
                    <div class="row">
                        <div class="col-md-12">
                            <h1>Register</h1>
                        </div>
                    </div>
                </div>
            </section>
        </header>
        <div class="container">
            <div class="row">
                <div class="col-md-4 col-md-offset-4">
                    <h2><strong>Register</strong></h2>
                    <hr/>
                    <?php
                        if($msg!="")
                        {
                            ?>
                            <div class="alert alert-success fade in">
                                <a href="#" data-dismiss="alert" class="close">&times;</a>
                                <?php echo $msg;?>
                            </div>
                            <?php

```

```

}
?>

```

```

<form method="post" action="" onSubmit="return validate_form();">
  <div class="row">
    <div class="form-group">
      <div class="col-md-12">
        <input type="text" name="fname" id="fname" value="<?php echo
$row->fname; ?>" class="form-control input-lg" placeholder="First Name"/>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="form-group">
      <div class="col-md-12">
        <input type="text" name="lname" id="lname" value="<?php echo
$row->lname; ?>" class="form-control input-lg" placeholder="Last Name"/>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="form-group">
      <div class="col-md-12">
        <input type="text" name="username" id="username" value="<?php
echo $row->username; ?>" class="form-control input-lg"
placeholder="Username"/>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="form-group">
      <div class="col-md-12">

```



```
        <input type="password" name="password" id="password"
value="<?php echo $row->password; ?>" class="form-control input-lg"
placeholder="Password"/>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
    <div class="form-group">
```

```
        <div class="col-md-12">
```

```
            <input type="text" name="phone" id="phone" value="<?php echo
$row->phone; ?>" class="form-control input-lg" placeholder="Phone Number"
size="10" maxlength="10"/>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
<div class="row">
```

```
    <div class="form-group">
```

```
        <div class="col-md-12">
```

```
            <input type="text" name="email" id="email" value="<?php echo
$row->email; ?>" class="form-control input-lg" placeholder="Email"/>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
<div class="row">
```

```
    <div class="form-group">
```

```
        <div class="col-md-12">
```

```
            <textarea name="address" id="address" value="<?php echo $row-
>address; ?>" class="form-control input-lg" placeholder="Address"></textarea>
```

```
        </div>
```

```
    </div>
```

```

</div>
<div class="row">
    <div class="form-group col-md-12">
        <?php
            $helper->getCoursesSelect();
        ?>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-12">
        <?php
            $helper->getCourseyearsSelect();
        ?>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-12">
        <?php
            $helper->getSemestersSelect();
        ?>
    </div>
</div>

<div class="row">
    <div class="form-group">
        <div class="col-md-12">
            <input type="submit" name="submit" class="btn btn-primary btn-lg"
value="Register now!"/>
        </div>
    </div>
</div>
</form>
</div>
<!-- Start Sidebar -->

```

```

        <aside class="col-md-3 sidebar right-sidebar">

            </aside>

        </div>

    </div>

</div>

</div>

<?php
    require_once "footer.php";
?>
</div>

<!-- End Body Container -->
<script src="js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->
<script src="plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="plugins/owl-carousel/js/owl.carousel.min.js"></script>
<script src="plugins/page-scroller/jquery.pagescroller.js"></script>
<script src="js/helper-plugins.js"></script> <!-- Plugins -->
<script src="js/bootstrap.js"></script> <!-- UI -->
<script src="js/init.js"></script> <!-- All Scripts -->
<!-- End Js -->
</body>
</html>

```

## References

### Books:-

- [1] “Web Programming”, by ‘Chris Bates’ Wiley Dreamtech India, 2<sup>nd</sup> Edition.
- [2] “Software Engineering”, Ian Somerville, Sixth Edition, Pearson Education Ltd.
- [3] “HTML Complete References” Easy steps to develop web pages.
- [4] “PHP Complete Reference”

### Websites:-

- [1] <http://en.wikipedia.org/wiki/PHP> for Php.
- [2] <http://www.hotscripts.com/category/php/> for Php
- [3] <http://www.mysql.com/click.php?e=35050> for MySQL.
- [4] <http://www.w3schools.com> for information on HTML.

# Appendices

## Chapter 1

1.1 Introduction	1
1.2 Scope of the caption project	2-3

## Chapter 2

2.1 Capstone project planning	
2.1.1 Work breakdown structure	4-5
2.1.2 Timeline developer-schedule	6-9
2.1.3 Cost breakdown structure	10-12
2.1.4 Capstone project risk assessment	12
2.2 Requirements specification	
2.2.1 Functional	13
2.2.2 Non-functional (Quality attributes)	14
2.2.3 User input	15
2.2.4 Technical constraints	16
2.3 Design specifications	
2.3.1 Chosen system design	17
2.3.2 Discussion of alternative design	18-20
2.3.3 Detailed description of components/subsystems	21-22
2.3.4 Components 1-n	22

## Chapter 3

3.1 Approach and methodology	
3.1.1 Discuss the technology	23-24
3.1.2 Methodologies	25

3.1.3 Use cases	26
3.1.4 Programming	27-28
3.1.5 Analysis	28
3.1.6 Process design	29-32
3.1.7 Product design	33-37
3.1.8 Fabrication	38-39

## **Chapter 4**

4.1 Test and validation	
4.1.1 Test plan	40-41
4.1.2 Test approach	42-43
4.1.3 Features tested	44
4.1.4 Features not tested	44
4.1.5 Findings	44-45
4.1.6 Inference	46-47

## **Chapter 5**

5.1 Business aspects	
5.1.1 The market and economic outlook of the project	48
5.1.2 Features	48
5.1.3 How does the project of service fit into the competitive landscape?	49
5.1.4 Who want the possible client or customer?	49-50
5.2 Financial considerations	
5.2.1 Capstone project budget	51
5.2.2 Cost of capstone projection needed for either profit or nonprofit option	51

5.3 Conclusion and recommendations	
5.3.1 State of completion of capstone project	52
5.3.2 Future work	52
<b>Program Code</b>	52-74
<b>References</b>	75
<b>Appendices</b>	76-80

## LIST OF FIGURES

<b>Figure number</b>	<b>Figure name</b>	<b>Page number</b>
Figure 2.3.1.1	System Architecture	17
Figure 2.3.2.1	Data flow diagram level 0	18
Figure 2.3.2.2	Data flow diagram level 1 admin	19
Figure 2.3.2.3	Data flow diagram level 1 user	20
Figure 2.3.3.1	Component diagram	21
Figure 3.1.2.1	Agile methodology diagram	25
Figure 3.1.3.1	Use case	26
Figure 3.1.6.1	List of tables	29
Figure 3.1.6.2	Admin table	29
Figure 3.1.6.3	Category table	30
Figure 3.1.6.4	Courses table	30
Figure 3.1.6.5	Members table	31
Figure 3.1.6.6	Questions table	31
Figure 3.1.6.7	Quiz result table	32
Figure 3.1.7.1	Index page	33
Figure 3.1.7.2	Dashboard	33
Figure 3.1.7.3	Add category	34
Figure 3.1.7.4	View category	34
Figure 3.1.7.5	Add question	35
Figure 3.1.7.6	View question	35
Figure 3.1.7.7	View result	36
Figure 3.1.7.8	Registration page	36
Figure 3.1.7.9	User login page	37
Figure 3.1.7.10	Test page	37



## LIST OF TABLES

<b>Table number</b>	<b>Table name</b>	<b>Page number</b>
Table 1.2.1	Cost of the project	3
Table 2.1.3.1	Final check	10
Table 2.1.3.2	Labor cost of work	11-12
Table 4.1.2.1	Admin login(test approach)	42
Table 4.1.2.2	Manage test	42
Table 4.1.2.3	Manage question and answer	42
Table 4.1.2.4	User registration	43
Table 4.1.2.5	User login	43
Table 4.1.2.6	Submit test	43
Table 5.2.2.1	Cost (profit/nonprofit)	51