

MINI PROJECT ON

ONLINE EXAMINATION SYSTEM

Submitted in partial fulfillment of 4th Semester

MASTER OF COMPUTER APPLICATIONS

Of

Visvesvaraya Technological University



Md Miraj Ansari
1AY18MCA14

Nitya Pandey
1AY18MCA17

Under the Guidance of

Prof. Venkatesh H



Acharya Institute of Technology

Acharya Dr.Sarvepalli Radhakrishna Road,
Soldevanahalli, Bangalore – 560090

www.acharya.ac.in

2020

ACHARYA INSTITUTE OF TECHNOLOGY
Department of MCA
Soladevanahalli, Hesaraghatta Main Road,
Bengaluru 560090.



C E R T I F I C A T E

This is to certify that the project entitled

Online Examination System
is a bonafide work carried out by

Md Miraj Ansari
1AY18MCA14

Nitya Pandey
1AY18MCA17

in partial fulfillment of 4th semester Master of Computer Applications
during the academic year 2019-20.

PROJECT GUIDE

Head of the Department

EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any work would be incomplete without mention of the people who made it possible, whose constant guidance and encouragement served as a beacon light and crowned our efforts with success.

We wish to thank our beloved principal Dr. **Prakash M.R** of Acharya Institute of Technology and management of Acharya Institutes.

We take this opportunity to thank Prof. **Manish Kumar Thakur**, Head of the Department of Master of Computer Applications, Acharya Institute of Technology and all the faculty members of MCA department, for the help extended throughout the tenure of our mini project work.

We are highly indebted to Prof. **Venkatesh H**, our project guide for taking keen interest in our work and her/his constant monitoring, invaluable guidance and support throughout the course of our project. We profusely thank her for having patience to clear our doubts and channelize our efforts. Her cheerful disposition made our work all the more enjoyable.

Last but not least, we also thank those who directly or indirectly involved in our mini project work.

Md Miraj Ansari

INDEX

PARTICULARS	PAGE NO
1.INTRODUCTION.....	01
2. SYSTEM ANALYSIS.....	02
2.1 OBJECTIVE	
2.2 SCOPE	
2.3 MODULES	
3. SOFTWARE REQUIREMENT SPECIFICATIONS.....	04
3.1 HARDWARE REQUIREMENTS	
3.2 SOFTWARE REQUIREMENTS	
3.3 FUNCTIONAL REQUIREMENTS	
3.4 NON-FUNCTIONAL REQUIREMENTS	
4. SYSTEM DESIGN.....	08
4.1 DATA FLOW DIAGRAM	
4.2 ER DIAGRAM	
5. SOURCE CODE.....	14
6. FEASIBILITY STUDY.....	19
7. IMPLEMENTATION.....	19
8. TESTING.....	20
8.1 BLACK BOX TESTING	
8.2 TEST CASES	
9. SCREENSHOTS.....	26
10. CONCLUSION.....	30
11. FUTURE ENHANCEMENTS.....	30
12. REFERENCES.....	30

1.INTRODUCTION

Online examinations are an important method of evaluating the success potential of students. This research effort the individuals under consideration were students who would be enrolling in computer courses or Technologies Registrations. A prototype of a web-based placement examination system is described from the stand point of the research effort, end user, and software development.

An on-line educational system including exam processing and electronic journal features. An instructor builds a course-based questions which on-line contain in identification of assignments. Which are compiled into an on-line exam syllabus?

Users enrolled in the platform may access the electronic details they provided and perform various functions with the on-line educational system in order to participate in the on-line examinations. Users can receive an on-line exam, having multimedia content, for the course, and they can electronically provide answers for the exam. And after Completion of their duration of exam they are provided the grade or marks secured in their examinations.

online examinations contents providers to focus on creating effective assessment questions and focusing on exam's feedback delivery to students. In the paper we present techniques that are pertinent to the elements of assessment process: answers submission, computerized grading, and feedback after submission. As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization.

The administrators, instructor, Students who are attending for online examination can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of Online Examinations like conducting Exams as per scheduled basis and delivering result to that particular use or student. And the details of students who attempted Online Examination are maintained at administrator.

2. SYSTEM ANALYSIS

2.1 OBJECTIVE

The objective of the Online Examination System is to provide better information for the users of this system for better results for their maintenance in student examination schedule details and grading details.

2.2 SCOPE

This application is used to conduct online examination. The students can sit at individual terminals and login to write the exam in the given duration. The questions have to be given to the students. This application will perform correction, display the result immediately and also store it in database. This application provides the administrator with a facility to add new exams. This application provides the Instructor to add questions to the exam, modify questions in the exam in a particular exam. This application takes care of authentication of the administrator, Instructor as well as the student.

2.3 MODULES

The mini project modules have been summarized as follows:

- **UserLogin**

In the user login a user visiting the website has away to login and access the full feature of the website and taking examination based on the subjected choices he/she optsfor.

- **Signing Up**

In the signing up pages when a new user visits the website, he/she has to create an id for the online examination system website so that they can access the content of the website. A new user coming for the first time has to create the id using a valid email address and password that should be eight mixed characters long.

- **Administrator login**

Online Examination System

Administrator login is the hidden panel in the website which can be accessed only by the committee that is holding the examination admin panel will provide with surplus features of adding, deleting questions updating questions and checking the score of the students taking examinations.

- **Multiple choice questions**

In this module the users will be able to take the examination and for every correct answer it will be a plus one marking with a wrong he/she gets a minus one.

- **Evaluation system**

The examination conducting committee using this website will be able to evaluate the merits of the students based on the marks they obtained in test

Advantages and Disadvantages

Advantages: Here are the advantages that have been kept in mind while making this project.

- The big benefit of online examination is the reduction of costs and time, both from the student as from the teacher.
- With online examination students can do the exam online, in their own time and with their own device, regardless where they live.
- The benefit of online based examination is that any institute can host exams in a more convenient way which is taking over the flaws in existing manual examination technique.

Disadvantages: Some drawbacks are always there which this website carries.

- It requires regular updation of the questions so as to keep the questions with current trend and topics.
- Every question cannot be kept up as multiple choices, some need to be summarized.
- It requires an active internet connection which should be readily available and all time connected until the exam gets over. Else there will be problem in evaluating the marks or even the test may fail.

3. SOFTWARE REQUIREMENT SPECIFICATIONS

A software requirements specification (SRS) is a detailed description of a software system to be developed with its function and non-functional requirements. The SRS is developed based on the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consists of all necessary requirements required for project development. To develop the software system, we should have clear understanding of Software system. To achieve this, we need to have continuous communication with customers to gather all requirements.

A good SRS defines how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real-life scenarios. Using the Software requirements specification (SRS) document, QA lead, managers create test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Let's see how to test SRS and the important points to keep in mind while testing it.

- **Correctness of SRS should be checked:** Since the whole testing phase is dependent on SRS, it is very important to check its correctness. There are some standards with which we can compare and verify.
- **Ambiguity should be avoided:** Sometimes in SRS, some words have more than one meaning and this might confuse testers making it difficult to get the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.
- **Requirements should be complete:** When tester writes test cases, what exactly is required from the application, is the first thing which needs to be clear. For e.g. if application needs to send the specific data of some specific size then it should be clearly mentioned in SRS that how much data and what is the size limit to send.

Consistent requirements: The SRS should be consistent within itself and consistent to its reference documents. If you call an input “Start and Stop” in one place, don’t call it “Start/Stop” in another. This sets the standard and should be followed throughout the testing phase.

- Verification of expected result: SRS should not have statements like “Work as expected”, it should be clearly stated that what is expected since different testers would have different thinking aspects and may draw different results from this statement.
- Testing environment: some applications need specific conditions to test and also a particular environment for accurate result. SRS should have clear documentation on what type of environment is needed to setup.
- Pre-conditions defined clearly: one of the most important part of test cases is pre-conditions. If they are not met properly then actual result will always be different expected result. Verify that in SRS, all the pre-conditions are mentioned clearly.
- Requirements ID: these are the base of test case template. Based on requirement Ids, test case ids are written. Also, requirements ids make it easy to categorize modules so just by looking at them, tester will know which module to refer. SRS must have them such as id defines a particular module.
- Security and Performance criteria : security is priority when a software is tested especially when it is built in such a way that it contains some crucial information when leaked can cause harm to business. Tester should check that all the security related requirements are properly defined and are clear to him. Also, when we talk about performance of a software, it plays a very important role in business so all the requirements related to performance must be clear to the tester and he must also know when and how much stress or load testing should be done to test the performance.
- Assumption should be avoided: sometimes when requirement is not cleared to tester, he tends to make some assumptions related to it, which is not a right way to do testing as assumptions could go wrong and hence, test results may vary. It is better to avoid assumptions and ask clients about all the “missing requirements” to have a better understanding of expected results.

- Deletion of irrelevant requirements : there are more than one team who work on SRS so it might be possible that some irrelevant requirements are included in SRS. Based on the understanding of the software, tester can find out which are these requirements and remove them to avoid confusions and reduce workload.
- Freeze requirements : when an ambiguous or incomplete requirement is sent to client to analyse and tester gets a reply, that requirement result will be updated in the next

SRS version and client will freeze that requirement. Freezing here means that result will not change again until and unless some major addition or modification is introduced in the software.

Most of the defects which we find during testing are because of either in complete requirements or ambiguity in SRS. To avoid such defects, it is very important to test software requirements specific ation before writing the test cases. Keep the latest version of SRS with you for reference and keep yourself updated with the latest change made to the SRS. Best practice is to go through the document very carefully and note down all the confusions, assumptions and incomplete requirements and then have a meeting with the client to get them clear before development phase starts as it becomes costly to fix the bugs after the software is developed. After all the requirements are cleared to a tester, it becomes easy for him to write effective test cases and accurate expected results.

3.1 HARDWARE REQUIREMENTS

Processor	Minimum Requirements Intel Core 2 Duo (2.2Ghz)
Ram	Minimum 1GB or Above
Hard disk	At least 20GB of space
Keyboard	Any PS/2 or USB Keyboard
Mouse	Any PS/2 or USB Mouse
Monitor	15" Monitor

3.2 SOFTWARE REQUIREMENTS

Front End	HTML, CSS, BOOTSTRAP
Back End	PHP, MYSQL
Operating System	Windows, Linux Or Any Mobile devices

3.3 FUNCTIONAL REQUIREMENTS

The main purpose of functional requirements is to define all activities or operation that take place in the system. These are derived through interaction with the users of the system. Since requirements specification is a comprehensive document and contains a lot of data, it has been broken down into different stages in this report.

3.4 NON-FUNCTIONAL REQUIREMENTS

Reliability: Reliability the correction of an item, scale or instruction with a hypothetical one, which truly measures what is supposed to. Since the true instruction is not available.

Usability: usability refers to the capability of the product to be understood, learned and user friendly to users, when used under specified conditions. This section should include all those requirements that affect usability.

Maintainability: Maintainability is the ease with which program/specification can be corrected if an error occurs due to a change in requirements. Specify attributes of software that relate to ease of maintenance of the software itself.

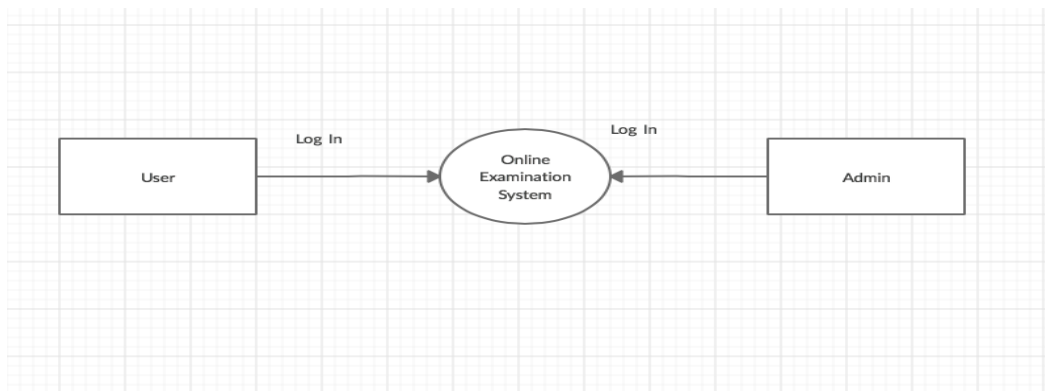
4. SYSTEM DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analysed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its state of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into the presentation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.1 DATAFLOW DIAGRAMS

Data objects represented by labelled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level. The DFD enables the software engineer to develop models of the information domain & functional domain at the same time.



4.2 DESIGN

A computer procedure is a series of operations designed to manipulate data to produce outputs from a computer system. The procedure may be a single program or a series of programs. The detail design of the computer procedure follows acceptance by management of an outline design proposal. The aim now is to design procedures at lower levels of detail, which will define the detailed steps to be taken to produce the specified computer output.

Design Tools

Various tools are being used by system analysis to specify computer procedures. Not all of them are used here to design this project. Some of the important tools that have been made use of are:

Entity relationship Diagram.

Input design.

Output Design.

Output Design.

Input design

Input design is a part of overall system design, which requires very careful attention. Often the collection of input data is the most expensive process of the system. In terms of both the equipment used and the number of people involved, it is the point of most contact for the users with the computer system; and it is prone to error. If data going into the system is incorrect, then the processing and output will magnify their errors.

One of the early activities of input design is to determine the nature of the input data. This is done partially in logical system design but it now needs to be made more explicit.

Error avoidance and Detection

Every effort must be made to ensure that input data remains accurate from the stage at which it is recorded and documented to the stage at which the customer accepts it. While every effort is

made to avoid errors during the preparation of input data, a proportion of errors are likely to be present.

Data validation

Computer input procedures must also be designed to detect errors in the data at a lower level of detail which is beyond the capability of the control procedures. These are combined with the design of the input process itself.

Output design

The specification of user requirements is the starting point for the appraisal and the detailed physical design must be done in the light of this and with continuing user involvement. The normal procedure is to design the outputs in detail first and then to work back to the inputs. The output can be in the form of operational documents, lengthy reports, and replies to queries or summarizing graphs.

Outputs from computer systems are required primarily to provide a permanent copy of the results for later consultation. Any data item not yet defined must be identified and recorded before output design can proceed. There is often a need at output to provide totals at various levels. It is not always desirable to print or display data as it is held on a computer. The system analyst must ensure whether the form in which it is stored in the system is suitable for the output. In proposed system the users have been provided with many outputs in the form of messages and alerts so as to help the user enter the correct data.

Reports

Reports enhance the application programmer's effort to output the formatted data in a manner practical for the user. This also helps to create hard copy of the valid information.

4.3 ENTITY RELATIONSHIP DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects.

Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.

It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the enduser.

In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Connectivity and Cardinality

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A one-to-one (1:1) relationship is when at most one instance of an entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A one-to-many (1:N) relationship is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity

A. An example of a 1: N relationship is a department has many employees each employee is assigned to one department

A many-to-many (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated

4.4 E-R NOTATION

There is no standard for representing data objects in ER diagrams. Each modelling methodology uses its own notation. The original notation used by Chen is widely used in academic texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

Online Examination System

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

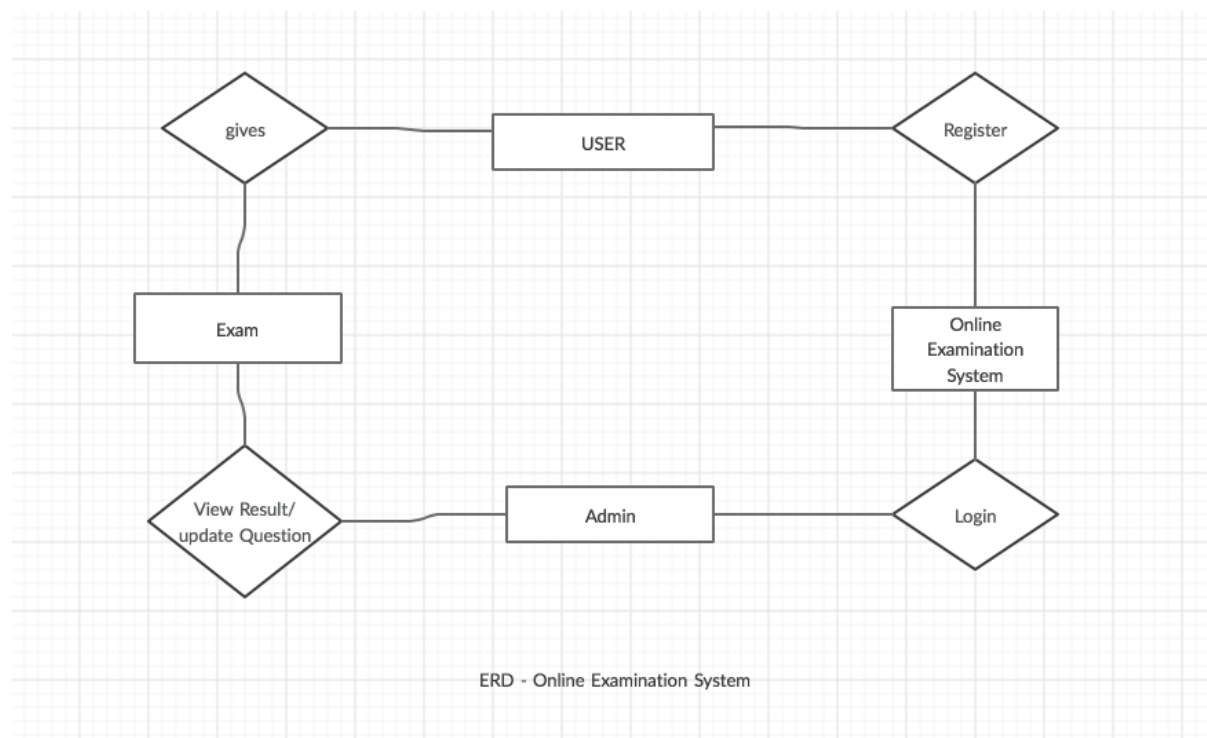
Entities are represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.

Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs

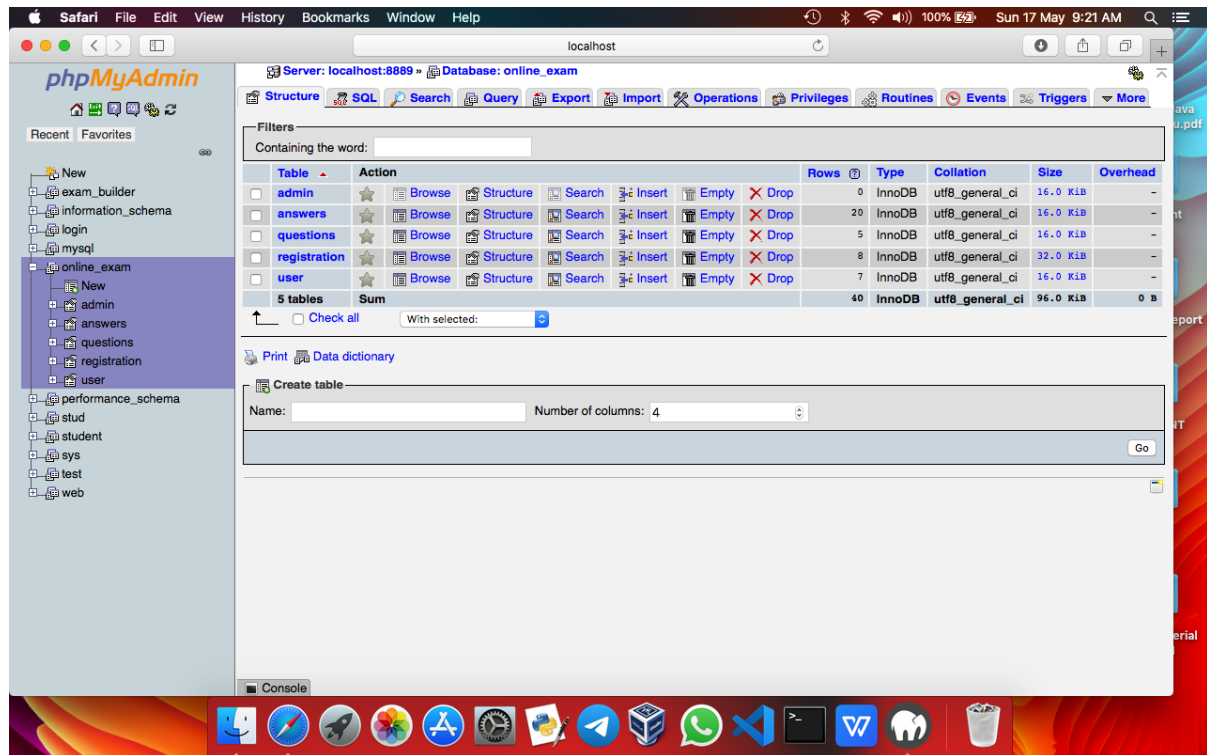
Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.

Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

4.5 ER – DIAGRAM



4.6 DATABASE TABLE



5. SOURCE CODE

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Online Examinaton in India | MyExam.com</title>

    </meta>

    <link rel="stylesheet" href="style.css">

    <link href="https://fonts.googleapis.com/css2?family=Bree+Serif&display=swap"
rel="stylesheet">

    <link
href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2:wght@500&family=Bree+Serif
&display=swap" rel="stylesheet">

    <link rel="stylesheet" media="screen and (max-width:1170px)" href="phone.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

    <nav id="navbar">

        <div id="logo">

        </div>

        <ul>

            <li class="item"><a href="#">Home</a></li>
```

```
<li class="item"><a href="#services-container">Examination</a></li>

<li class="item"><a href="register.php">Registration</a></li>

<li class="item"><a href="#contact-box">Contact Us</a></li>

</ul>

</nav>

<section id="home">

<h1 class="h-primary">Welcome to Online Examination</h1>

<p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Animi,
beatae quis! Consequuntur hic ab totam itaque earum repellendus error
accusantium, veritatis, beatae enim nihil atque provident a. Corporis,
ipsam temporibus.</p>

<center><button class="btn"><a href="login.php">Login</a></button></center>

</section>

<section id="services-container">

<h1 class="h-primary center">Examination</h1>

<div class="services">

<div class="box">



<h2 class="h-secondary center">Online Examinaton</h2>

<p class="center"> Lorem ipsum dolor sit amet consectetur adipisicing elit.

Unde sit ipsam eius, assumenda tempore et animi nulla odit pariatur recusandae vero
autem labore,

.</p>

</div>

<div class="box">
```


<h2 class="h-secondary center">Online Examinaton</h2>

<p class="center"> Lorem ipsum dolor sit amet consectetur adipisicing elit.

Unde sit ipsam eius, assumenda tempore et animi nulla odit pariatum recusandae vero autem labore,

.</p>

</div>

<div class="box">

<h2 class="h-secondary center">Online Examinaton</h2>

<p class="center"> Lorem ipsum dolor sit amet consectetur adipisicing elit.

Unde sit ipsam eius, assumenda tempore et animi nulla odit pariatum recusandae vero autem labore

</p>

</div>

</div>

</section>

<!-- Sponsor -->

<center>

<div class="col-sm-4">

<h2>Connect</h2>


```
</div>

</center>

<!-- Contact Section -->

<section id="contact">

  <h1 class="h-primary center">Contact Us</h1>

  <div id="contact-box">

    <form action="">

      <div class="form-group">

        <label for="name">Name</label>

        <input type="text" name="name" id="name" placeholder="Enter your name">

      </div>

      <div class="form-group">

        <label for="name">Email Id</label>

        <input type="email" name="name" id="email" placeholder="Enter your email">

      </div>

      <div class="form-group">

        <label for="name">Phone</label>

        <input type="phone" name="name" id="phone" placeholder="Enter your phone
number">

      </div>

      <div class="form-group">

        <label for="name">Message</label>

        <textarea name="message" id="message" cols="34" rows="10"></textarea>

      </div>

    </form>
```

</div>

</section>

<footer>

<div class="center">

Copyright © www.OnlineExam.com. All right reserved

</div>

</footer>

</body>

</html>

6. FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a highlevel capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility.

Technical Feasibility The project entitles "Courier Service System" is technically feasibility because of the below mentioned feature. The project was developed in PHP It provides the high level of reliability, availability and compatibility. All these make PHP an appropriate language for this project. Thus, the existing software PHP is a powerful language.

Economic Feasibility The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

Operational Feasibility In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessities.

7. IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

8. TESTING

The testing phase is an important part of software development. It is the pauperized system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Software testing is carried out in three steps:

1.The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So, errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So, unit testing is conducted to individual modules.

2.The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may result in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.

3.The final step involves validation and testing which determines which the software functions as the user expected. Here also some modifications were. In the completion of the project it is satisfied fully by the end user.

This section deals with the details of the classes of tests which must be conducted to validate the functions, performance, and the constraints. This is achieved basically by the means of testing which plays a vital role in the development of the software. The various low-level testing which can be grouped on a broader sense are discussed as below:

Unit Testing: Testing of each and every module individually is called unit testing. Each module is designed and executed so that it works independently and generates some relative output. Unit testing of a module results in a set of individually executing modules that generates its

independent output. The modules in this project i.e. purchases, sales, returns, cash and bank transactions, stock status and others are individually tested.

Integration Testing: Integration testing is one in which the modules are combined with one another to synchronize the outputs and co-ordinate the flow of control. Each module is coalesced with other module to establish proper connectivity between the two modules. All the individual modules have been integrated and tested.

Validation Testing: Validation testing deals with testing that whether the validations are justified and all the requirements are fulfilled logically as per the requirements as the validations are designed according to the requirements of the application – an approach to the problem in a logical and relational manner. All the validation testing have been done (like only the numbers should be accepted in the phone number, amount, bill no fields, quantity and other related fields, similarly for the text to accepted only in related fields like the names and other related fields). The stock validation is the major validation that keeps track of the products dealt and the reorder levels.

System Testing: Testing the whole system i.e. verifying and validating the whole source code to check if the system as a whole is executing is called system testing. The proposed system undergoes it under all conditions to be called a successful system.

8.1 Black Box Testing

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance. The higher the level, and hence the bigger and more

User input validation

User input must be validated to conform to expected values. For example, if the software program is requesting input on the price of an item, and is expecting a value such as 3.99, the software must check to make sure all invalid cases are handled. A user could enter the price as "-1" and achieve results contrary to the design of the program. Other examples of entries that could be entered and cause a failure in the software include: "1.20.35", "ABC", "0.000001", and "9999999999". These are possible test scenarios that should be entered for each point of user input.

Other domains, such as text input, need to restrict the length of the characters that can be entered. If a program allocates 30 characters of memory space for a name, and the user enters 50 characters, buffer overflow condition can occur.

Typically, when invalid user input occurs, the program will either correct it automatically, or display a message to the user that their input needs to be corrected before proceeding.

complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

Data validation

Computer input procedures must also be designed to detect errors in the data at a lower level of detail which is beyond the capability of the control procedures. These are combined with the design of the input process itself.

Online Examination System

The validation procedure must be designed to check each record, data item, field against certain criteria specified by the system analyst or the programmer. Some Examples of data validation done in this project are:

Invalid loginid cannot be entered

Invalid Address, such as door no, etc(advance>balance or negative) cannot be entered.

8.2 TEST CASES

Test case id	Test cases	Preconditions	Input test data	Expected o/p	Pass	Fail	Assumed O/p
1	Test if user is able to login successfully.	User must be registered already	correct username, correct password	User must successfully login to the web page	Y	N	Login successful
2	Test if unregistered users is not able to login to the site		incorrect username, incorrect password	Please enter the correct username and password for a staff account. Note that both fields may be case- sensitive.	Y	N	Please enter the correct username and password for staff account. Note that both fields may be case- sensitive
3	Test with valid username and empty password such that login must get failed	User must be registered already	valid username and empty password	Please correct the error below. This field is required.	Y	N	Please correct the error below. This field is required.
4	Test with empty username and valid password such that login must get failed	registered user's password	empty username and valid password	Please correct the errors below. This field is required.	y	N	

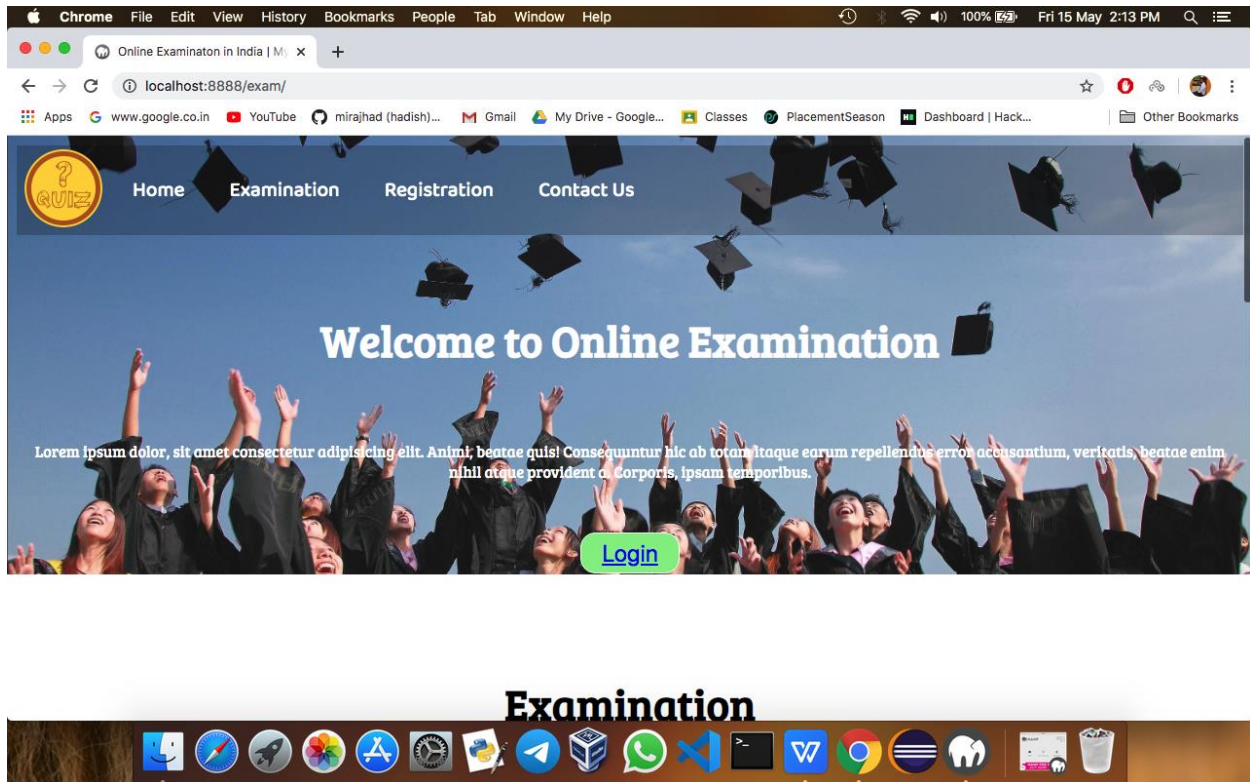
Online Examination System

5	Test with empty username and empty password and check if login fails	-	-	Please correct the errors below. This field is required. This field is required.	N	Y	Please Fill the field
---	--	---	---	---	---	---	-----------------------

9. SCREENSHOTS

HOMEPAGE

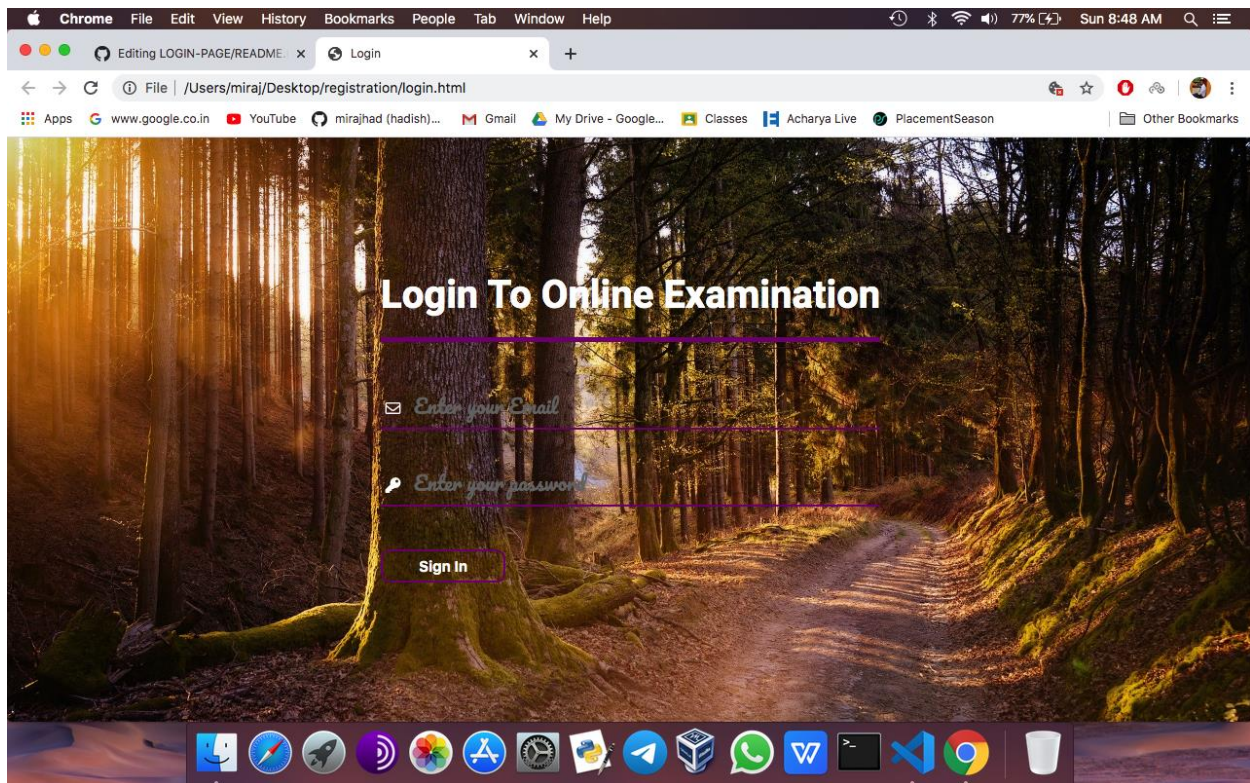
DATE:-15/04/20



Online Examination System

USER LOGIN

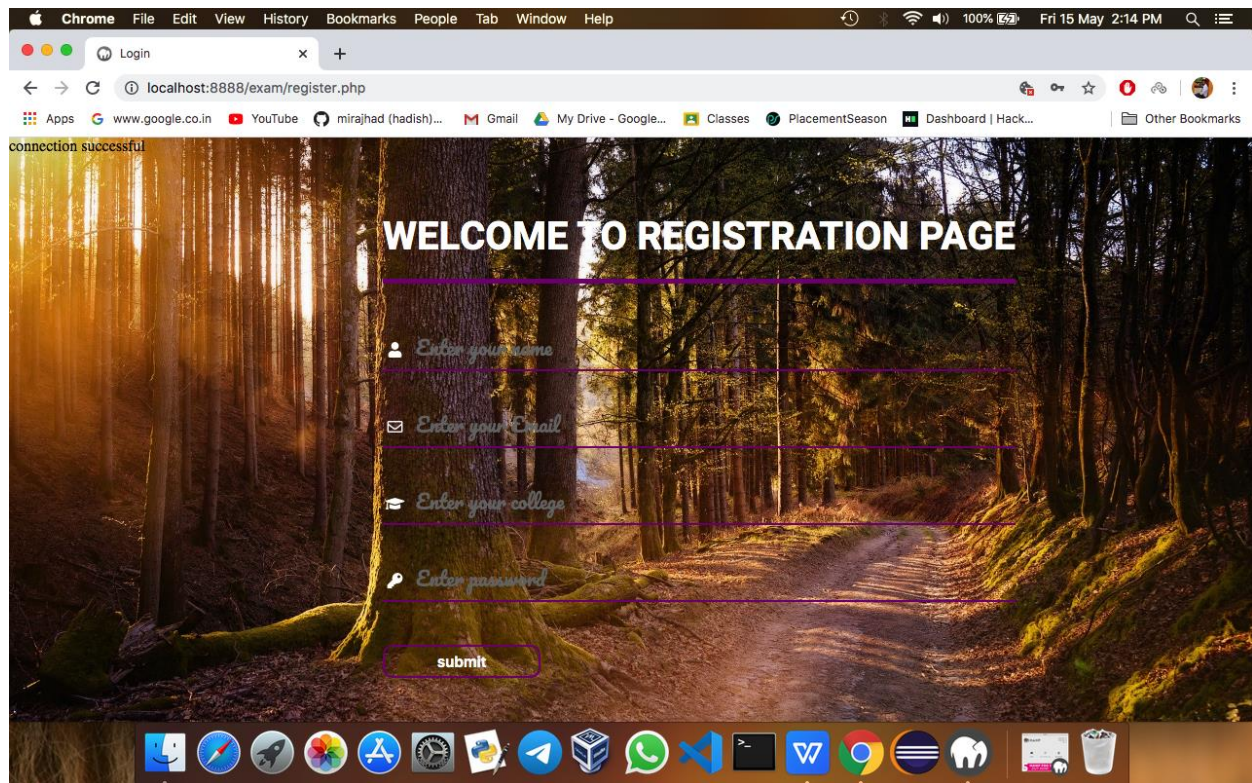
DATE:-17/04/20



Online Examination System

REGISTRATION

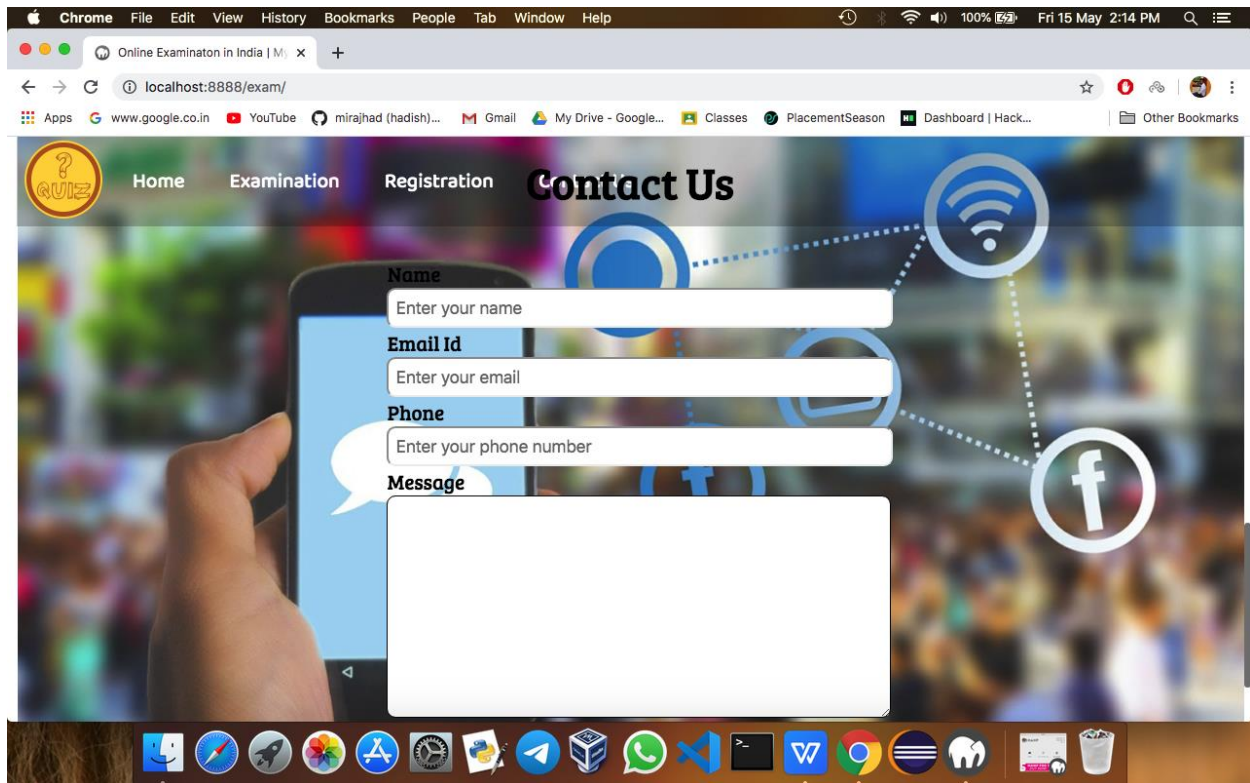
DATE:-25/04/20



Online Examination System

FEEDBACK

DATE:-16/04/20



10. CONCLUSION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

Automation of the entire system improves the efficiency

- 1.It provides a friendly graphical user interface which proves to be better when compared to the existing system.
2. It gives appropriate access to the authorized users depending on their permissions.
3. It effectively overcomes the delay in communications.
4. Updating of information becomes so easier.
5. System security, data security and reliability are the striking features.
6. The System has adequate scope for modification in future if it is necessary.

11. FUTURE ENHANCEMENTS

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the different scheduled examinations information that are Currently issued. Well I and my team members have worked hard in order to present an improved website better than the existing one's regarding the information about the various activities. Still, we found out that the project can be done in a better way. Primarily, when we request information about a particular schedule it just shows the exam date and platform. So, after getting the information we can get access to the online exam. The enhancement that we can add the searching option. We can directly search to the particular student details from this site.

12. REFERENCES

- 1.Codewithharry.com
- 2.Jp web
- 3.Slideshare.com

