

**A MAJOR PROJECT REPORT
ON
CERTIFICATE
GENERATOR**

A dissertation submitted in partial
fulfilment of requirements for awarding
the degree of

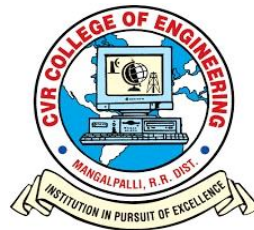
**Bachelor of Technology
in
INFORMATION TECHNOLOGY**

Submitted by

**T Sai Charan – (18B81A1288)
P Sai Charan Reddy – (18B81A1289)
P Yashwanth Reddy– (18B81A12B9)**

Under the esteemed guidance of

D Bhanu Mahesh
Asst. Professor, IT Department
CVR College of Engineering



**DEPARTMENT OF INFORMATION TECHNOLOGY
Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510
2021-2022**



Cherabuddi Education Society's
CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : <http://cvr.ac.in>, email : info@cvr.ac.in

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

CERTIFICATE

This is to certify that the Project Report entitled “**Certificate Generator**” is a bonafide work done and submitted by **T Sai Charan (18B81A1288)**, **P Sai Charan Reddy (18B81A1289)** and **P Yashwanth Reddy (18B81A12B9)** during the academic year 2021-2022, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

Project Guide

Mr B Bhanu Mahesh
Assistant Professor
Information Technology

Head of Department

Dr. Bipin Bihari Jayasingh
Professor
Information Technology

Project Coordinator

Dr. J. Sengathir
Associate Professor,
Information Technology

External Examiner

DECLARATION

We hereby declare that the project report entitled “**Certificate Generator**” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in Information Technology and it is a record of bonafide project work carried out by us under the guidance of D Bhanu Mahesh, Asst. Professor, Department of Information Technology. We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

Signature of Student
T Sai Charan
(18B81A1288)

Signature of Student
P Sai Charan Reddy
(18B81A1289)

Signature of Student
P Yashwanth Reddy
(18B81A12B9)

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

We offer our sincere gratitude to our internal guide, D Bhanu Mahesh, Asst. Professor of IT Department, for his immense support, timely co-operation, and valuable advice throughout the course of our project work.

We would like to thank the Head of Department, Professor Dr. Bipin Bihari Jayasingh, for his meticulous care and cooperation throughout the project work. We are thankful to Dr. J. Sengathir, Project Coordinator, Associate Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances. We also thank the Project Review Committee Members for their valuable suggestions.

ABSTRACT

Online contest has taken a huge peak in recent covid days. Most of the organizations are preferring to conduct their competitions and exams online. Also conducting online contest have been difficult these days if one wants to distribute certificates to participants.

This portal ensures to conduct contest to each and every student along with generating certificates. These certificates can be generated directly by student when ever they complete their contest successfully. Not only students can generate certificates but also after concluding of contest faculty can also directly send certificates to the students registered emails at one go.

	<u>LIST OF TABLES</u>	Page No.
5	Testing	
5.1	Test Cases	71

	<u>LIST OF FIGURES</u>	Page No.
4	Design	
4.1	Use-Case Diagram	17
4.2	Sequence Diagram	18
4.3	ER Diagram	19
5	Implementation	
	Landing page or Home page	74
	Student Registration Page	74
	Student Verification Page	75
	Student forgot password Page	75
	Student Login Page	76
	Faculty Registration Page	76
	Faculty Verification Page	77
	Faculty Forgot Password Page	77
	Faculty Login Page	78
	Faculty Dashboard Page	78
	Faculty Creating New Contest Page	79
	Faculty Check History of students who attempted contest	79
	Student Dashboard Page	80
	Student Attempting Contest	80
	Student checking their certificate	81
	Downloaded certificate	82

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
1	INTRODUCTION 1.1 Literature Survey	8 9
2	SOFTWARE REQUIREMENT SPECIFICATIONS 2.1 Functional Requirements 2.2 Non-Functional Requirements	10 10 12
3	DESIGN 3.1 Use Case Diagram 3.2 Sequence Diagram 3.3 ER Diagram	14 17 18 19
4	IMPLEMENTATION 4.1 Introduction 4.2 Modules	20 20 22
5	TESTING	67
6	CONCLUSION Future Enhancements References Appendix A: Abbreviations Appendix B: Software installation process Appendix C: Software Usage Process	83 84 85 86 87 91

Chapter-1

INTRODUCTION

1.1 Introduction

The web services these days are being used on almost every sector and aspect of our day – to – day life. Educational Sector is also one such major aspect where Internet and Web Services can impact the lives of students on a larger scale.

The web application has become a public square, a library, a doctor's office, a shop, a school, a design studio, an office, a cinema, a bank, and so much more. With proper design and implementation, we have built a web application where:

- The web application works as an automatic distributor for distributing the certificates regarding the contests hosted by that particular user or Organizer.
- It is delivered to the end user through the use of computers using standard Internet technology.
- It focuses on simpler work process than the traditional distribution/assignment paradigms.

This project is aimed to give a platform to the end user overcome the difficulty of certifying each and every participant/contestant. Sometimes, the result of participants may vary from one participant to another based on their performance in a particular contest. In such cases, the certifying process may become even more difficult. In such scenarios, this project would more helpful in terms of reducing work load.

The main features that are provided by the application include:

1. Easy Workflow.
2. Accessible from any type of device such as Mobile, Laptop.

3. User Data is secured and consistent.
4. Simple UI.
5. Saves time and effort

1.2 Literature Survey

1. HackerRank:

- <https://www.hackerrank.com/administration/contests/create>
- Hackerrank is a most popular website we all come across. Hackerrank allows us to host our contents online

2. Google Docs AutoCrat add-on:

- After successful completion of contests using hackerrank, Hackerrank generates an excel sheet with the participant details and their results.
- Using this excel sheets, we can generate certificate with our certificate template and send certificates to participants to their mail id's and also store these certificates in our local storage.

1.3 Objective

As the Project – “Get Certified” is aimed to deliver an efficient way of certifying and sending the certificate to the participant, the Objectives of the project rely upon the factors that abide to give the best User experience and immense pleasure to the user of the web application.

The main objectives of Get Certified are:

1. Support the contest organizers to enhance the level of hosting.
2. Track all the information and details of contests.
3. Track all the information of the participants of a particular contests.
4. Ensures better distribution of results or outputs.
5. Providing all the functionality related to events and tracks all the information and details of events.
6. Providing Quality-based features to the users.

Chapter-2

Software Requirements Specifications

2.1 Introduction

A Software Requirements Specification (SRS) is a description of a particular software product, program or set of programs that performs a set of functions in a target environment (IEEE Std. 830-1993).

2.2 Functional Requirement Specification

2.2.1 User requirement

This project is developed using graphics in Nodemon which is a tool that helps develop Node. js based applications by automatically restarting the node application when file changes in the directory are detected. The options available are displayed in a menu format, like in an online editor. Clicking on any particular menu item through mouse or through keyboard a dropdown menu is displayed, listing all the options available under that menu item and the user can select the needed actions according to their wish.

3.2.2 Software requirement

Operating System	:	Windows XP/7
Front End	:	HTML, CSS
Scripts	:	Nodemon (JavaScript).
Database	:	MongoDB

3.2.3 Hardware requirement

- Hard disk : 80 GB
- RAM : 1 GB
- Processor : i3

Feasibility Study

Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

Operational Feasibility

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project? Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project. Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

Technical Feasibility

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, .at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies.

Non- Functional Requirements

The major non-functional Requirements of the system are as follows

- Usability

The system is designed with completely automated process hence there is no or less user intervention.

- Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

- Performance

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will

give response to the end user on client system with in very less time.

- Supportability

The system is designed to be the cross platform supportable.

The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

- Implementation

The system is implemented with Java environment. The java software development kit and net beans used as software and windows xp professional is used as the platform.

Chapter 3

DESIGN

3.1 Introduction

Design is a meaningful engineering representation of something that is to be built. Software design is a process through which the requirements are translated into a representation of the software. Design is the place where quality is fostered in software engineering. Design is the perfect way to accurately translate a customer's requirement into a finished software product. Design creates a representation or model, provides detail about software data structure, architecture, interfaces and components that are necessary to implement a system. This chapter discusses about the design part of the project. Here in this document the various UML diagrams that are used for the implementation of the project are discussed.

Design Principle

The Unified Modelling Language (UML) is a visual modelling language used to specify, visualize, construct and document a software intensive system. The embedded real-time software systems encountered in applications such as telecommunications, school systems, aerospace, and defines typically tends to be large and extremely complex. It is crucial in such systems that the software is designed with a sound architecture. A good architecture not only simplifies construction of the initial system, but also, readily accommodates changes forced by a steady stream of new requirements.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the

design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML are: Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts. Be independent of particular programming languages and development processes. Provide a formal basis for understanding the modelling language. Encourage the growth of the OO tools market. Support higher-level development concepts such as collaborations, frameworks, patterns and components. Integrate best practices.

3.2 UML Description

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

- User Model View
 - This view represents the system from the user's perspective.
 - The analysis representation describes a usage scenario from the end-user's perspective.
- Structural model view
 - In this model the data and functionality are arrived from inside the system.
 - This model view models the static structures.
- Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- Implementation Model View

In this the structural and behavioural as parts of the system are represented as they are to be built.

- Environmental Model View

In these the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modelling, this focuses on the user model and structural model views of the system.
- UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

UML Diagrams

USE CASE DIAGRAM

Use Case: Use case describes the behaviour of a system. It is used to structure things in a model. It contains multiple scenarios, each of which describes a sequence of actions that is clear enough for outsiders to understand.

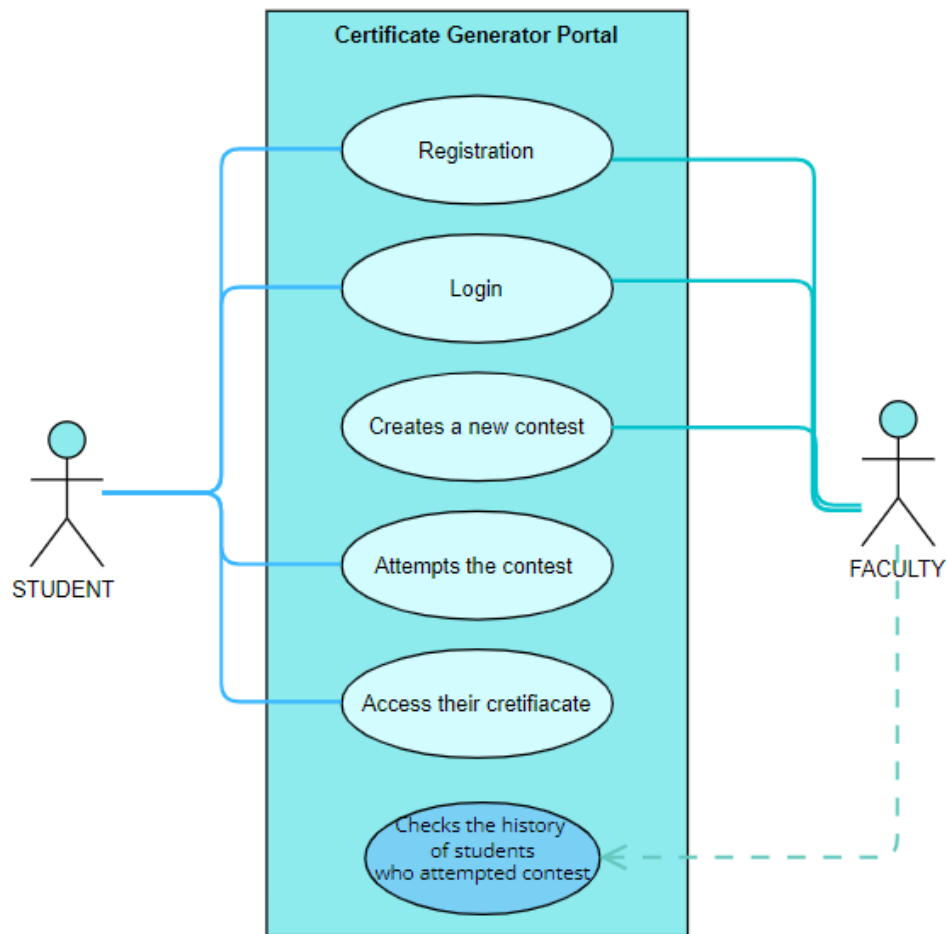


Fig 1: Use Case Diagram

SEQUENCE DIAGRAM:

This diagram is simple and visually logical, so it is easy to see the sequence of the flow of control. It also clearly shows concurrent processes and activations in a design.

Object: Object can be viewed as an entity at a particular point in time with a specific value and as a holder of identity that has different values over time. Associations among objects are not shown. When you place an object tag in the design area, a lifeline is automatically drawn and attached to that object tag.

Actor: An actor represents a coherent set of roles that users of a system play when interacting with the use cases of the system. An actor participates in use cases to

accomplish an overall purpose. An actor can represent the role of a human, a device, or any other systems.

Message: A message is a sending of a signal from one sender object to other receiver object(s). It can also be the call of an operation on receiver object by caller object. The arrow can be labeled with the name of the message (operation or signal) and its argument values

Duration Message: A message that indicates an action will cause transition from one state to another state.

Self Message: A message that indicates an action will perform at a particular state and stay there.

Create Message: A message that indicates an action that will perform between two states.

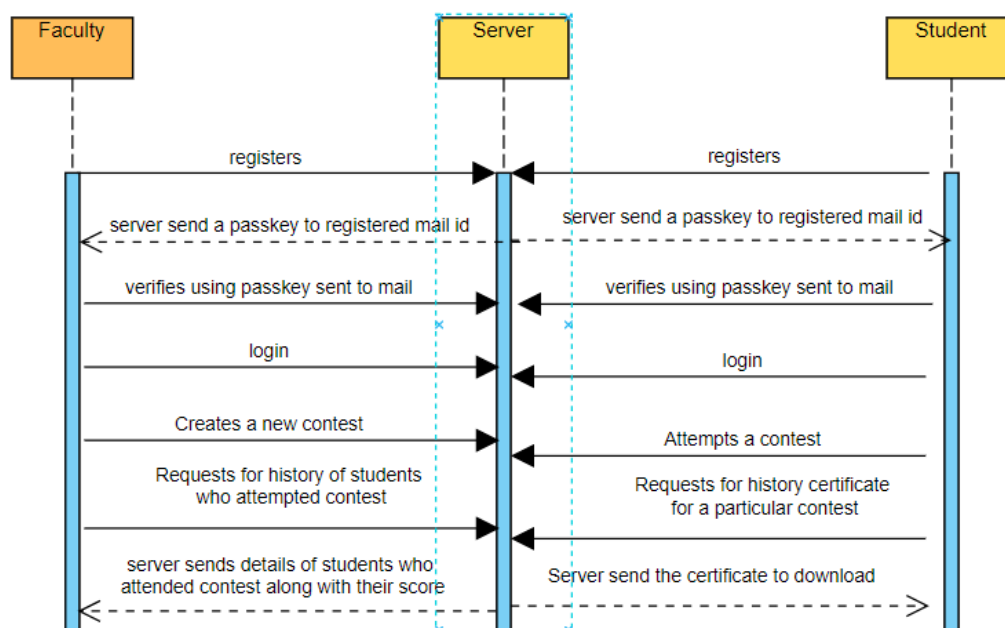


Fig-2: Sequence Diagram

ER Diagram:

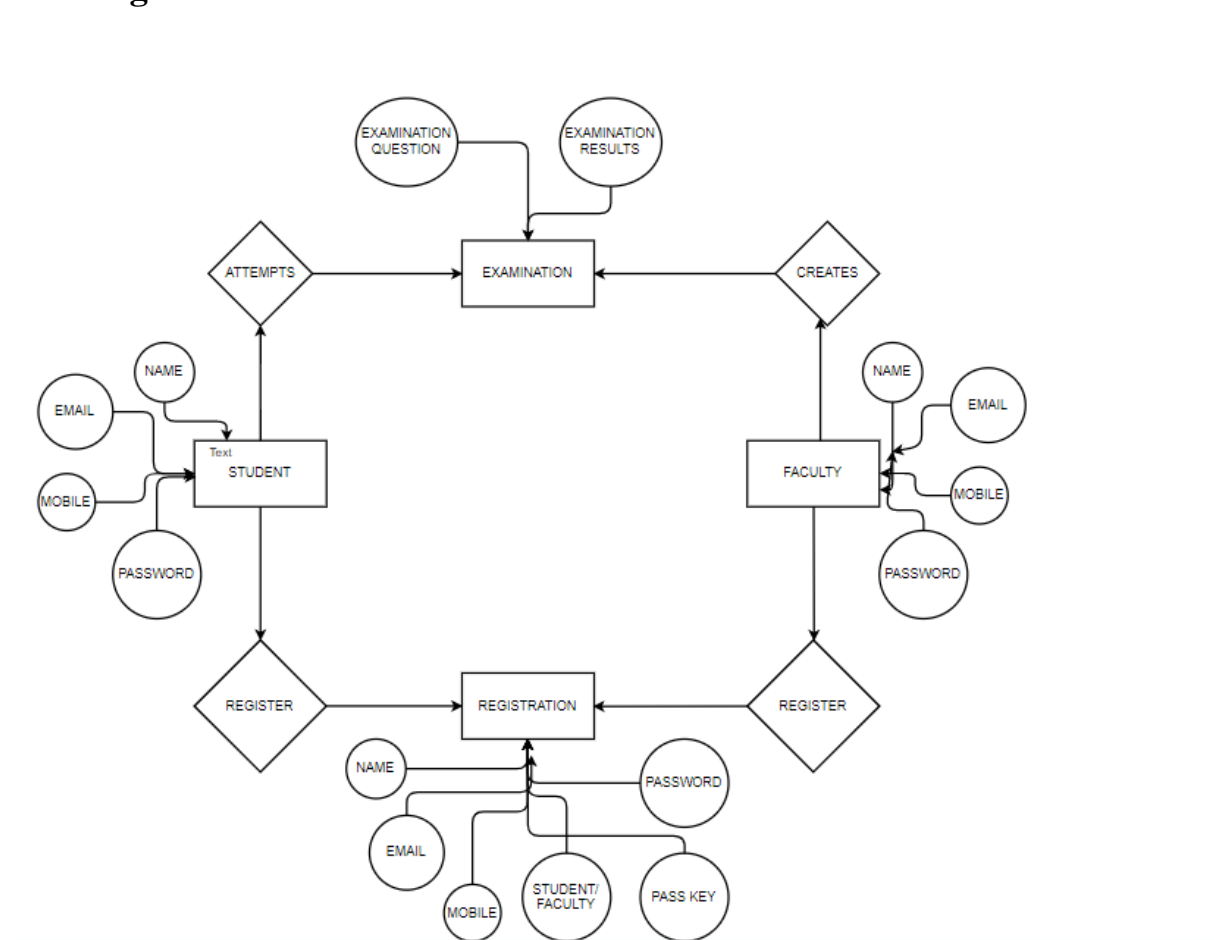


Fig-3: ER Diagram

Chapter 4

IMPLEMENTATION

4.1 Introduction

The most crucial phase of any project is the implementation. This includes all those activities that take place to convert from the old system to the new system. It involves setting up of the system for use by the concerned end user. A successful implementation involves a high level of interaction between the analyst, programmers and the end user. The most common method of implementation is the phased approach, which involves installation of the system concurrently with the existing system. This has its advantage in that the normal activity carried out, as part of the existing system is anyway hampered. The end users are provided with sufficient documentation and adequate training in the form of demonstration/presentation to familiarize with the system.

4.2 Explanation of Key functions

Step 1: Student and faculty verification: Any user who are registers first time will be given a uniquely generated hash code which will be sent to their provided email. Only after verification of their passkeys they can access the portal

Step 2: Faculty creating a new contest: Any verified faculty can be able to create a contest at any time they want. To create a new contest a faculty first need to navigate to faculty dashboard, there they can find a button “Add Contest”. By pressing this button they can create an new contest by entering the required fields.

Step 3: Faculty accessing the participation history of a contest: A faculty can access the history of students who have attempted any particular contest. See participation is a button that was provided corresponding to each and every contest. By pressing that button any faculty can view the participants of any contest that they have created.

Step 4: Student attending a contest: A student can find all the active contests that they can attend from a student dashboard. Any student can attend a contest only once. In order to generate a certificate a student first need to successfully attempt a contest.

Step 5: Student generating their certificate: After completion of certificate student can view as well as download their certificate.

4.3Code snippets of project:

- 1) Nodemon server to create a server that binds all the html and js files and starts a server:

```
var express = require('express');
var app = express();
var mongoose= require('mongoose');
var passport = require('passport');
const sgMail = require("@sendgrid/mail");
var expressFileUpload = require('express-fileupload')
const config = require("./backend/config");
sgMail.setApiKey(config.SendgridAPIKey);
var path = require('path');
var ejs = require('ejs')
var apis= require('./backend/api/allapiroutes.js');
var uis= require('./backend/ui/alluiroutes.js');

app.use(express.urlencoded({ extended: true }));

app.use(express.json());

app.use(expressFileUpload())
```

```

// app.set('views',path.join(__dirname,'views'))
// app.set('view engine','ejs')
// app.use(express.static(__dirname + '/views'));

app.use((req, res, next) => {
    res.header('Access-Control-Allow-Origin', '*');
    next();
});

//DataBase Connection
var Connection_String=config.dbURI;
var options={useUnifiedTopology: true, useNewUrlParser: true };
mongoose.connect(Connection_String,options);
mongoose.connection.on('connected', function()
{ console.log("GetCertified DataBase Connected");})

// var userauth= require("./backend/api/userauth");
// app.use("/api/auth",userauth);

app.use(express.static(__dirname+'/frontend'));

app.use('/api',apis);
app.use('/',uis);

var port= process.env.PORT || 3000;

```

```

app.listen(port,()=>
{ console.log("http://localhost:"+port)
});

```

```

app.use((error, req, res, next) => {
  res.status(error.status || 500);
  res.json({
    error: {
      message: error.message,
    },
  });
});

```

2) Landing page of project:

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

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

```
<meta name="viewport" content="width=device-width, initial-
scale=1">
```

```
<!-- The above 3 meta tags *must* come first in the head; any other
head content must come *after* these tags -->
```

```
<title>Get Certified</title>
```

```
<!-- Latest compiled and minified CSS -->
```

```

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css" integrity="sha512-
iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ymDWZCvqsWgccV
3gFoRBv0z+8dLJgyAHlR35VZc2oM/gI1w=="
crossorigin="anonymous" referrerpolicy="no-referrer" />

```

```

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min
.css" integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTs
z/K68vbdEjh4u" crossorigin="anonymous">

```

```

    <link rel="stylesheet" href="css/index.css">

```

```

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements
and media queries -->

```

```

    <!-- WARNING: Respond.js doesn't work if you view the page via
file:// -->

```

```

    <!--[if lt IE 9]>

```

```

        <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script
>

```

```

        <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

```

```

    <![endif]-->

```

```

</head>

```

```

<body data-spy="scroll" data-target="#myScrollspy">

```

```

    <nav class="navbar navbar-default navbar-fixed-top" data-spy="affix"
data-offset-top="205" style="padding:8px">

```

```

        <div class="container-fluid">

```

```

            <div class="navbar-header">

```

```

                <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#myNavbar">

```

```

                    <span class="icon-bar"></span>

```

```

                    <span class="icon-bar"></span>

```

```

                    <span class="icon-bar"></span>

```

```

                </button>

```

```

                <a class="navbar-brand" href="#">

```

```

                    <span> 
</span>Get certified

```



```

        </a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#about">About</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </div>
</div>
</nav>

<header style="background-
image:url('https://i.pinimg.com/originals/8c/6e/ea/8c6eea90524b4d992db
e84c23a36bf47.gif');">
    <div class="header-content">
        <div class="header-content-inner">
            <h1 style="font-size: 5em;">Get certified</h1>
            <p>provide the report and leave us the rest we are here to
handle </p>
            <a href="/accessStudent" class="btn btn-default-new">Get
Started as Student</a> </div>
            <a href="/accessFaculty" class="btn btn-default-new">Get
Started as Faculty</a> </div>
        </div>
    </div>
</header>

<section class="about-section bg-color" id="about">
    <div class="container">
        <div class="row">
            <div class="col-md-12 col-sm-12 col-xs-12">
                <h3>About</h3>

```

<p>Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications. Contestants are referred to as sport programmers. Competitive programming is recognized and supported by several multinational software and Internet companies, such as Google and Facebook. There are several organizations who host programming competitions on a regular basis. A programming competition generally involves the host presenting a set of logical or mathematical problems, also known as puzzles, to the contestants (who can vary in number from tens to several thousands), and contestants are required to write computer programs capable of solving each problem. Judging is based mostly upon number of problems solved and time spent for writing successful solutions, but may also include other factors (quality of output produced, execution time, program size, etc.) At the end, all these sport programmers get certification based on their performance and excellence. It is practically difficult to certify each programmer individually. So “Get-Certified” is the web application which issues certification to all sport participants through online at a time, directly to their mail id’s.</p>

</div>

</div>

</section>

<footer id="contact">

<div class="container">

<div class="row">

<div class="col-md-8 col-md-offset-2 col-sm-12 col-xs-12">

<h3>Let's Get In Touch!</h3>

<p>Ready to start your next project with us? That's great! Give us a call or send us an email and we will get back to you as soon as possible!</p>

<div class="gap">

<div class="col-sm-6 col-xs-12">

123-456-789

</div>

<div class="col-sm-6 col-xs-12">

```
<span class="glyphicon glyphicon-envelope"></span>  
  
    <a  
href="mailto:get@bootstrap.com">get@bootstrap.com</a>  
  
    </div>  
  
    </div>  
  
    </div>  
  
    </div>  
  
    </div>  
  
</footer>  
  
    <iframe      title="practice1"      width="1140"      height="541.25"  
src="https://app.powerbi.com/reportEmbed?reportId=fe426c82-5465-  
42e2-8606-cf258d305140&autoAuth=true&ctid=da464968-10f5-4b7b-  
8217-  
d6ab822e3949&config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly93YWJp  
LWluZGhlLWNlbnRyYWwtYS1wcmltYXJ5LXJlZGlzZWNOUmFuYWw  
x5c2lzLndpbmRvd3MubmV0LyJ9"      frameborder="0"  
allowFullScreen="true"></iframe>  
  
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->  
  
    <script   src="https://use.fontawesome.com/releases/v5.15.3/js/all.js"  
data-auto-replace-svg="nest"></script>  
  
    <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"><  
/script>  
  
    <!-- Include all compiled plugins (below), or include individual files as  
needed -->  
  
    <!-- Latest compiled and minified JavaScript -->  
  
    <script src="/js/index.js"></script>  
  
    <script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js  
"      integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD  
8wGNICPD7Txa" crossorigin="anonymous"></script>  
  
</body>
```

</html>

3) Signup page:

```
function signup() {  
    //Email verification  
    function IsEmail(email) {  
        var regex = /[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!  
9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9]*[a-z0-9])?;/;  
        if (!regex.test(email)) return false;  
        else return true;  
    }  
  
    //Phone Number verification  
    function IsPhoneno(phoneno) {  
        var regex = /^[7-9][0-9]{9}$/g;  
        if (!regex.test(phoneno)) return false;  
        else return true;  
    }  
    function Ispassword(password)  
    {  
        var paswd = /^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!  
9!@#$%^&*]{7,15}$/;  
        if(password.match(paswd)) return true;  
        else return false;  
    }  
  
    //console.log("Hello");  
    var emailid =  
String(document.getElementsByClassName("emailid")[0].value);  
    var phoneno =  
String(document.getElementsByClassName("phone")[0].value);
```

```

var password =
String(document.getElementsByClassName("pswrd")[0].value);

var conformpassword=
String(document.getElementsByClassName("cnfrmpswrd")[0].value);

var name =
String(document.getElementsByClassName("name")[0].value);

//alert(emailid+ " " +phoneno+password+" " +conformpassword+" "
+name);

var c = 6;
if (name == "") {

    document.getElementById("alert").innerHTML = `<div class="alert
alert-danger alert-dismissible">

    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

    <strong>Sorry!</strong>Please Enter your name

</div>`;

    c--;
}
else if (emailid == "") {

    document.getElementById("alert").innerHTML = `<div class="alert
alert-danger alert-dismissible">

    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

    <strong>Sorry!</strong>Please Enter the Email

</div>`;

    c--;
}
else if(password!=conformpassword)
{

```

```

        document.getElementById("alert").innerHTML = `<div class="alert
alert-danger alert-dismissible">

        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

        <strong>Sorry!</strong>Passwords doesnot match

</div>`;

        c--;

    }

    else if (password == "") {

        document.getElementById("alert").innerHTML = `<div class="alert
alert-danger alert-dismissible">

        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

        <strong>Sorry!</strong>Please Enter the password

</div>`;

        c--;

    }

    else if(password.length<8)

    {

        document.getElementById("alert").innerHTML = `<div class="alert
alert-danger alert-dismissible">

        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

        <strong>Sorry!</strong>Password should contain atleast 8 charectes

</div>`;

        c--;

    }

    if (c == 6) {

        // alert("Hello");

```

```

    if (!IsEmail(emailid)) {
        document.getElementById("alert").innerHTML = `<div
class="alert alert-danger alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Sorry!</strong>Invalid Email
        </div>`;
        c--;
    }
    else if (!IsPhoneno(phoneno) && phoneno != "") {
        document.getElementById("alert").innerHTML = `<div
class="alert alert-danger alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Sorry!</strong>Invalid phone No
        </div>`;
        c--;
    }
    else if (!Ispassword(password)) {
        //alert("Hello");
        swal({
            title: "Sorry!",
            text: "password should contain Minimum eight characters, at
least one uppercase letter, one lowercase letter and one number",
            type: "warning",
            timer: 6000
        });
        c--;
    }
}

```

```

//ajax call to create an instance to the user in database
if (c == 6) {

$.ajax({
    type: "POST",
    url: "/api/user/signup",
    async: false,
    data: {
        email: emailid,
        name: name,
        mobileNumber: phoneno,
        password: password
    },
    success: function(resultData) {
        if (resultData.message == "Email already exists")
            document.getElementById("alert").innerHTML = `<div
class="alert alert-danger alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                <strong>Sorry!</strong>Enter the Email
            </div>`;
        if (resultData.message == "user created") {
            window.location.href = '/verify';
        }
    }, //sucess
    error: function(resultData) {
        if (resultData.responseJSON.message == "Unauthorized
access") {
            location.href = "/"

```



```

    } else {
        var x = document.getElementById("alert");

        x.innerHTML = `<div class="alert alert-danger alert-
dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Sorry!</strong>Enter the Email
        ${resultData.responseJSON.message}</div>`
        x.className = "show";
        setTimeout(function() { x.className =
x.className.replace("show", ""); }, 3000);
    }
} //error
});
}
} //End of signup function

```

4) Certificate template:

```

var templates = {
    simple : {
        style : '@page { size: landscape; }\
        #org{ text-align: center;margin-top: 35px;font-family:
"Arima Madurai"; }\
        #head{ text-align: center;font-family: "Arvo";margin-
top: 10px;font-size: 60px; }\
        #sub-head{ text-align: center;margin-top: 30px;font-
family: "Barlow Semi Condensed"; }\
        #name{ text-align: center;margin-top: 20px;font-family:
"Bai Jamjuree";text-decoration: underline; }\
        #remarks{ text-align: center;margin-top: 30px;font-
family: "Baloo 2"; }\
        #issued{ font-family: "Baloo 2";margin-left:
15px;margin-bottom: 15px; }\
        .credits{ font-family: "Barlow";font-size: 18px;margin-
bottom: 50px; }',

```

```

main : '<div class="container" style="border: 13px solid
rgb(144, 201, 255);">\
    <h3 id="org"></h3>\
    <h1 id="head"></h1>\
    <h4 id="sub-head">This Certificate is presented
to</h4>\
    <h2 id="name"></h2>\
    <h4 id="remarks"></h4>\
    <h6>Issued on : </h6><h6 id="issued"></h6>\
    <div class="d-flex justify-content-around"
style="margin-left: 25px;">\
        <div>Issued By: <h3 align="center" class="credits"
id="issuer"></h3>\
            </div>\
        <div>Hosted By: <h3 align="center" class="credits"
id="host"></h3>\
            </div>\
        <div>Sponsored By: <h3 align="center"
class="credits" id="sponsor"></h3>\
            </div>\
        </div>\
    </div>'
},

```

```

simpleTwo : {
    style : '@page { size: landscape; }\
    .tag-center{\
        height: 50px;\
        width: 80px;\
        background-color: #ffd000;\
        margin-left: 20px;\
    }\
    .tag-left{\
        width: 0;\
        height: 0;\
        border-left: 0px solid transparent;\
        border-right: 40px solid transparent;\
        border-top: 50px solid #ffd000;\
        margin-left: 20px;\
    }\
    .tag-right{\
        width: 0;\
        height: 0;\

```

```

border-left: 40px solid transparent;\
border-right: 0px solid transparent;\
border-top: 50px solid #ffd000;\
}\
#org{ text-align: center;font-family: "Arima Madurai"; }\
#head{ text-align: center;font-family: "Arvo";margin-
top: 10px;font-size: 60px; }\
#sub-head{ text-align: center;margin-top: 30px;font-
family: "Barlow Semi Condensed"; }\
#name{ text-align: center;margin-top: 20px;font-family:
"Bai Jamjuree";text-decoration: underline; }\
#remarks{ text-align: center;margin-top: 30px;font-
family: "Baloo 2"; }\
#issued{ font-family: "Baloo 2";margin-left:
15px;margin-bottom: 15px; }\
.credits{ font-family: "Barlow";font-size: 18px;margin-
bottom: 50px; }',
main : '<div class="container" style="border: 15px solid
rgb(164, 144, 255);background-color: rgb(252, 253, 255);">\
<div class="tag-center"></div>\
<div class="d-flex justify-content">\
<div class="tag-left"></div>\
<div class="tag-right"></div>\
</div>\
<h3 id="org"></h3>\
<h1 id="head"></h1>\
<h4 id="sub-head">This Certificate is presented
to</h4>\
<h2 id="name"></h2>\
<h4 id="remarks">\
<p> --remarks-- </p>\
</h4>\
<h6>Issued on : </h6><h6 id="issued"></h6>\
<div class="d-flex justify-content-around"
style="margin-left: 25px;">\
<div>Issued By: <h3 align="center" class="credits"
id="issuer"></h3>\
</div>\
<div>Hosted By: <h3 align="center" class="credits"
id="host"></h3>\
</div>\
<div>Sponsored By: <h3 align="center"
class="credits" id="sponsor"></h3>\

```

```

        </div>\
    </div>\
</div>'
    }
}

module.exports = templates

```

5) contest.js:

```

const mongoose = require("mongoose");
const User= require("../user");

const contestSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  contestname : { type: String, required: true },
  userId: { type: mongoose.Schema.Types.ObjectId, ref: "User" },
  userName: { type: String, required: true },
  participants:[{
    userId: { type: mongoose.Schema.Types.ObjectId, ref: "User" },
    score:{type: Number}
  }],
  num_questions: { type: Number},
  questions:[{question:{type: String},answer:{type:String},options:[]}],
  cert_id: { type: String, required: true },
  creationtime: { type: Number},
  isDeleted: { type: Boolean, default: false },
  organisation: { type: String, required: true },
  description: { type: String, required: true }
});

```

```
module.exports = mongoose.model("Contest", contestSchema);
```

6) participant.js:

```
const mongoose = require("mongoose");
```

```
const User= require("./user");
```

```
const Contest= require("./contest");
```

```
const participantSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  contestName : { type: String, required: true },
  ContestId: { type: mongoose.Schema.Types.ObjectId, ref: "Contest" },
  name: { type: String},
  email: {type: String,
    lowercase: true,
    match: /[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~
  ]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/,
  },
  rank: { type: Number },
  certified: { type: Boolean, default: false },
  emailsent: { type: Boolean, default: false },
  passkey: {type:String,required: true },
  isDeleted: {type:String, default: false }
});
```

```
module.exports = mongoose.model("Participant", participantSchema);
```

7) user.js:

```
const mongoose = require("mongoose");
```

```
const Contest = require("./contest");
```

```

const Participant=require('./participants');

const userSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  googleId: {type: String},
  name: { type: String },
  userType: { type: String, default: "User" },
  email: {
    type: String,
    lowercase: true,
    match: /[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/,
  },
  mobileNumber: {
    type: Number,
    match: /^[7-9][0-9]{9}$/g,
  },
  password: { type: String},
  contests: [{
    contestId: { type: mongoose.Schema.Types.ObjectId, ref: "Contest" },
    score:{ type: Number },
  } ],
  token: {
    type: String,
  },
  passResetKey: { type: String },

```

```

    passKeyExpires: { type: Number },
    verificationKey: { type: String },
    verificationKeyExpires: { type: Number },
    isEmailVerified: { type: Boolean, default: false },
    isDeleted: { type: Boolean, default: false },
    creationtime: {type: Number}
  });

```

```

module.exports = mongoose.model("User", userSchema);

```

8) user_auth.js:

```

const express = require("express");
const mongoose = require("mongoose");
const jwt = require("jsonwebtoken");
const bcrypt = require("bcrypt");
const shortid = require("shortid");
const passport = require("passport");
var GoogleStrategy = require( 'passport-google-oauth2' ).Strategy;
const User = require("../models/user");
const router = express.Router();
var item=require("../itemlib");

```

```

passport.use(new GoogleStrategy({
  clientID: "31856671834-
lcgthafc6opobvo6f8tq1u6ung0stjb0.apps.googleusercontent.com",
  clientSecret: "ifCYYCOgh3VMPK9UJMWelrhk",
  callbackURL: "/api/auth/google/redirect",

```

```

    },
    function(request, accessToken, refreshToken, profile, done) {
        console.log(profile.id);
    }
));

router.get("/google", passport.authenticate("google", { scope: ["profile",
"email"], }));

//Callback route for google to redirect
router.get("/google/redirect", (req, res, next) => {
    console.log(req);
    res.redirect("/");
}
);

```

```

module.exports = router;

```

9) itemlib.js:

```

exports.getAllItems = function(itemModel, cb) {
    console.log('Getting All Items');
    var query = { }; // get all
    itemModel.find(query, function(err, allDBItems) {
        cb(err, allDBItems);
    });
};

```



```

// populateJson = { path: 'fans', select: 'name' }
exports.getAllItemsWithPopulate = function(itemModel, populateJson, cb) {
  console.log('Getting All Items With Populate');
  var query = {}; // get all
  itemModel
    .find(query)
    .populate(populateJson)
    .exec(function(err, allDBItems) {
      cb(err, allDBItems);
    });
};

exports.getItemById = function(id, itemModel, cb) {
  console.log('Getting Single item with ID ' + id);
  itemModel.findById(id, function(err, singleDBItem) {
    cb(err, singleDBItem);
  });
};

exports.getItemByIdWithPopulate = function(id, itemModel, populateJson, cb)
{
  console.log('Getting Single item by ID ' + id + ' with populate');
  itemModel
    .findById(id)
    .populate(populateJson)
    .exec(function(err, singleDBItem) {
      cb(err, singleDBItem);
    });
};

```

```

    });
};

exports.getItemByQuery = function(query, itemModel, cb) {
    // console.log('Getting item with Query ' + JSON.stringify(query));
    itemModel.find(query, function(err, allDBItems) {
        if (err) console.log('ERROR: ' + err);
        cb(err, allDBItems);
    });
};

```

```

exports.getItemByQueryWithPopulate = function(query, itemModel,
populateJson, cb) {
    console.log('Getting item with Query ' + JSON.stringify(query) + ' with
populate');
    itemModel.find(query).populate(populateJson).exec(function(err,
allDBItems) {
        if (err) console.log('ERROR: ' + err);
        cb(err, allDBItems);
    })
};

```

```

exports.getItemByQueryWithSelect = function(query, itemModel, selectJson,
cb) {
    console.log('Getting item with Query ' + JSON.stringify(query) + ' with
select');
    itemModel.find(query).select(selectJson).exec(function(err, Allitems) {
        if (err) console.log('ERROR: ' + err);
        cb(err, Allitems);
    });
};

```

```
    })  
  }  
}
```

```
exports.getItemByQueryWithPopulateAndSelect = function(query, itemModel,  
populateJson, selectJson, cb) {
```

```
  console.log('Getting item with Query ' + JSON.stringify(query) + ' with  
populate and select');
```

```
  itemModel.find(query)
```

```
    .populate(populateJson)
```

```
    .select(selectJson)
```

```
    .exec(function(err, Allitems) {
```

```
      if (err) console.log('ERROR: ' + err);
```

```
      cb(err, Allitems);
```

```
    })
```

```
  }
```

```
exports.createOrSkipByQuery = function(query, itemModel, itemDetails, cb) {
```

```
  console.log('Getting Single item with Query ' + JSON.stringify(query));
```

```
  itemModel.findOne(query, function(err, singleItem) {
```

```
    if (err) console.log('ERROR: ' + err);
```

```
    if (singleItem) {
```

```
      cb({ message: 'Skipping as this already exists' }, singleItem);
```

```
    } else {
```

```
      // INSERT
```

```
      exports.createitem(itemDetails, itemModel, cb);
```

```
    }
```

```
  });
```

```
};
```

```

exports.createOrUpdateByQuery = function(query, itemModel, itemDetails, cb)
{
  console.log('Getting Single item with Query ' + JSON.stringify(query));
  itemModel.findOne(query, function(err, singleItem) {
    if (err) console.log('ERROR: ' + err);
    if (singleItem) {
      // UPDATE
      itemDetails._id = singleItem._id;
      itemDetails.updated_at = new Date();
      exports.updateItem(itemDetails, itemModel, cb);
    } else {
      // INSERT
      exports.createitem(itemDetails, itemModel, cb);
    }
  });
};

```

```

exports.getSingleItemByQuery = function(query, itemModel, cb) {
  console.log('Getting Single item with Query ' + JSON.stringify(query));
  itemModel.findOne(query, function(err, singleItem) {
    if (err) console.log('ERROR: ' + err);
    cb(err, singleItem);
  });
};

```

```

exports.getSingleItemByQueryAndSortedOnField = function(
  query,

```

```

    fieldName,
    itemModel,
    cb
  ) {
    console.log('Getting Single item with Query ' + JSON.stringify(query));
    itemModel
      .findOne(query)
      .sort(fieldName)
      .exec(function(err, singleItem) {
        if (err) console.log('ERROR: ' + err);
        cb(err, singleItem);
      });
  };

```

```

exports.createitem = function(itemDetails, itemModel, cb) {
  console.log('Create New item for ' + JSON.stringify(itemDetails));
  var ti = new itemModel(itemDetails);
  ti.save(function(err) {
    if (err) console.log('ERROR ' + err);
    cb(err, ti);
  });
};

```

```

exports.createManyItems = function(itemDetails, itemModel, cb) {
  console.log("Inserting multiple items...");
  itemModel.insertMany(itemDetails, (err, data) => {
    if (err) console.log('ERROR ' + err);
  });
};

```

```

        cb(err, data);
    })
}

exports.updateItem = function(itemDetails, itemModel, cb) {
    console.log('Edit Resource ' + itemDetails._id);
    //console.log("MODEL: " + JSON.stringify(itemModel))
    itemModel.findById(itemDetails._id, function(err, qObj) {
        if (err || !qObj) cb(err, null);
        else {
            if (itemDetails._id) delete itemDetails._id;

            console.log(JSON.stringify(itemDetails));
            for (var p in itemDetails) {
                //console.log(itemDetails[p])
                if (itemDetails[p]) qObj[p] = itemDetails[p];
            }

            // Save Updated Statement
            qObj.save(function(err) {
                cb(err, qObj);
            });
        }
    });
};

exports.updateItemField = function(query, updateDetails, itemModel, cb) {

```

```

    console.log('updating details...');
    console.log(itemModel)
    itemModel.updateOne(query, updateDetails, (err, itemDetails) => {
        if (err) console.log('ERROR: ' + err);
        cb(err, itemDetails);
    });
};

exports.deleteMultipleItems = function(query, itemModel, cb) {
    console.log('Delete multiple resuorces which match ..' + query);
    itemModel.update(query, { "$set": { "isDeleted": true } }, { "multi": true },
    (err, details) => {
        if (err) console.log('ERROR: ' + err);
        cb(err, details);
    })
}

exports.deleteItem = function(id, softDelete, itemModel, cb) {
    console.log('Delete Resource ' + id);
    // cb(null, null); // Disabled Delete

    if (!softDelete) {
        itemModel.findByIdAndDelete(id, (err, details) => {
            if (err) console.log('ERROR: ' + err);
            cb(err, details);
        });
    } else {
        itemModel.findById(id, function(err, qObj) {
            if (err)

```

```

        cb(err, null);
    else {
        qObj.isDeleted = true;
        // Save Updated Statement
        qObj.save(function(err) {
            cb(err, qObj);
        });
    }
});
}

};

```

10) allroutes.js:

```

var express = require('express');
const app = express();
var path = require('path');

app.use(express.urlencoded({ extended: true }));
app.use(express.json());
//app.use(express.static(path.join(__dirname+'../../frontend')));

app.get("/", (req,res)=>
{
    var p=path.join(__dirname+'../../frontend/html/index.html')
    res.sendFile(p);})

```



```

app.get("/index", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/index.html')
        res.sendFile(p);})

app.get("/accessStudent", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/accessStudent.html')
        res.sendFile(p);})

app.get("/accessFaculty", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/accessFaculty.html')
        res.sendFile(p);})

app.get("/dashboardStudent", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/dashboardStudent.html')
        res.sendFile(p);})

app.get("/dashboardFaculty", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/dashboardFaculty.html')
        res.sendFile(p);})

app.get("/contest/:contestid", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/contest.html')
        res.sendFile(p);})

app.get("/Quiz/:contestid", (req,res)=>
    {   var p=path.join(__dirname+'../../frontend/html/dyn_quiz.html')
        res.sendFile(p);})

// app.get("/testingq/:contestid", (req,res)=>{
//     var p = path.join(__dirname+'../../frontend/html/dyn_quiz.html')

```

```

//    res.sendFile(p);
// })

app.get("/verify", (req,res)=>
{   var p=path.join(__dirname+'../../frontend/html/verify.html')
    res.sendFile(p);})

app.get("/frgtpswrd", (req,res)=>
{   var p=path.join(__dirname+'../../frontend/html/frgtpswrd.html')
    res.sendFile(p);})

app.get("/example", (req,res)=>
{   var p=path.join(__dirname+'../../frontend/html/example.html')
    res.sendFile(p);})

// app.get("/:contestid/:participantid", (req,res)=>
// {   var p=path.join(__dirname+'../../frontend/html/certificate.html')
//     res.sendFile(p);})

app.get("/results/:contestid", (req,res)=>
{   var p=path.join(__dirname+'../../frontend/html/results.html')
    res.sendFile(p);})

app.get("/getScores/:contestid", (req,res)=>
{   var p=path.join(__dirname+'../../frontend/html/Studentresults.html')
    res.sendFile(p);})

```

```
// app.get("/:contestid/:participantid",(req,res)=>{
//   res.render('certificate.ejs')
// })
```

module.exports=app;

11) verify.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Email verification</title>
  <link rel="stylesheet" href="/css/access.css">
  <link rel='stylesheet'
href='https://cdn.jsdelivr.net/npm/sweetalert2@7.12.15/dist/sweetalert2.min.css'
></link>
  <link
href="https://fonts.googleapis.com/css?family=Montserrat:400,700|Raleway:300,600" rel="stylesheet">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
integrity="undefined" crossorigin="anonymous">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></sc
ript>

<meta charset="UTF-8" />

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

<title>verify Email</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

<body>

<section style="padding-top: 3cm;">

<div class="container" style="border-radius: 30px;">

<div class="user signinBx" >

<div class="imgBx"></div>

<div class="formBx">

<div class="form">

<h2>Verify Your Email</h2>

<div id="alertmsg">

</div>

<input type="text" name="email" placeholder="Email"
class="email"/>

<input type="password" name="Passkey" placeholder="Passkey"
class="passkey"/>

<input type="submit" name="" value="Submit" onclick="verify()"
style="border-radius: 5px;"/>

<p class="signup">

<button style="background-color:white;border: none ;
color:#677eff ;height: 1cm; padding-left: 1cm;padding-right: 1cm; float: right;"
onclick="sendkey()" ><b><u>Resend verification key</u></b></button>

```

```

        </p>
    </div>
</div>
</div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/sweetalert2@7.12.15/dist/sweetalert2.all.min.js"></script>
    <script src="/js/verify.js"></script>
</body>
</html>

```

12) quiz.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contest Participants</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" integrity="sha512-iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8dLJgyAHlR35VZc2oM/gI1w==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

```

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/2.1.4/toastr.min.css"
integrity="sha512-
6S2HWzVFxruDlZxI3sXOZZ4/eJ8AcxkQH1+JjSe/ONCEqR9L4Ysq5JdT5ipqt
zU7WHalNwzwBv+iE51gNHJNqQ==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script
>
```

```
<link rel="stylesheet" type="text/css" href="../css/index.css">
```

```
<link rel="stylesheet" type="text/css" href="../css/contest.css">
```

```
<link rel="stylesheet" type="text/css" href="../css/quiz.css">
```

```
</head>
```

```
<body data-spy="scroll" data-target="#myScrollspy" style="background-color:
rgb(255, 253, 251);">
```

```
<nav class="navbar navbar-default navbar-fixed-top" data-spy="affix" data-
offset-top="205" style="padding:8px">
```

```
<div class="container-fluid">
```

```
<div class="navbar-header">
```

```
<button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#myNavbar">
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
</button>
```

```
<a class="navbar-brand" href="#">
```

` Get
certified`

``

`</div>`

`<div class="collapse navbar-collapse" id="myNavbar">`

`<ul class="nav navbar-nav navbar-right" >`

``

`Logout`

``

`</div>`

`</div>`

`</nav>`

`<main>`

`<div class="container">`

`<!--QUIZ -->`

`<div class="row">`

`<!-- Question 1 -->`

`<div class="col">

`

`Question 1: Number of primitive data types in Java are

`

`<div class="row pt-1">`

`<div class="col-md-3">`

`<div class="options"> <label class="option">6 <input type="radio"
value="a" id="que1" name="que1">
</label> </div>`

`</div>`

`<div class="col-md-3 mt-md-0 mt-3">`

```
        <div class="options"> <label class="option">7 <input type="radio"
value="b" id="que1" name="que1"> <span class="checkmark"></span>
</label> </div>
```

```
    </div>
```

```
    <div class="col-md-3 mt-md-0 mt-3">
```

```
        <div class="options"> <label class="option">8 <input type="radio"
value="c" id="que1" name="que1"> <span class="checkmark"></span>
</label> </div>
```

```
    </div>
```

```
    <div class="col-md-3 mt-md-0 mt-3">
```

```
        <div class="options"> <label class="option">9 <input type="radio"
value="d" id="que1" name="que1"> <span class="checkmark"></span>
</label> </div>
```

```
    </div>
```

```
    </div>
```

```
    </div>
```

```
</div>
```

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

```
    <!-- Question 2 -->
```

```
    <div class="col"><br><br>
```

```
        <span> Question 2: What is the size of float and double in java?</span>
<br><br>
```

```
    <div class="row pt-1">
```

```
        <div class="col-md-3">
```

```
            <div class="options"> <label class="option">32 & 64 <input
type="radio" value="a" id="que2" name="que2"> <span
class="checkmark"></span> </label> </div>
```

```
        </div>
```

```
    </div>
```



```
<div class="options"> <label class="option">32 & 32 <input
type="radio" value="b" id="que2" name="que2"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">64 & 64 <input
type="radio" value="c" id="que2" name="que2"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">64 & 32 <input
type="radio" value="d" id="que2" name="que2"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 3 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 3: Automatic type conversion is possible in which of
the possible cases?</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">byte to int <input
type="radio" value="a" id="que3" name="que3"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">int to long <input
type="radio" value="b" id="que3" name="que3"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">long to int <input
type="radio" value="c" id="que3" name="que3"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">short to int <input
type="radio" value="d" id="que3" name="que3"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 4 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 4: Select the valid statement.</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">char[] ch = new
char(5) <input type="radio" value="a" id="que4" name="que4"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```

        <div class="options"> <label class="option">char[] ch = new
char[5] <input type="radio" value="b" id="que4" name="que4"> <span
class="checkmark"></span> </label> </div>

    </div>

    <div class="col-md-3 mt-md-0 mt-3">

        <div class="options"> <label class="option">char[] ch = new char()
<input type="radio" value="c" id="que4" name="que4"> <span
class="checkmark"></span> </label> </div>

    </div>

    <div class="col-md-3 mt-md-0 mt-3">

        <div class="options"> <label class="option">char[] ch = new char[]
<input type="radio" value="d" id="que4" name="que4"> <span
class="checkmark"></span> </label> </div>

    </div>

</div>

</div>

</div>

</div>

<div class="row">

    <!-- Question 5 -->

    <div class="col"><br><br>

        <span> Question 5: When an array is passed to a method, what does the
method receive?</span> <br><br>

    </div class="row pt-1">

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

            <div class="options"> <label class="option">The reference of the
array <input type="radio" value="a" id="que5" name="que5"> <span
class="checkmark"></span> </label> </div>

        </div>

    </div class="col-md-3 mt-md-0 mt-3">

```

```
<div class="options"> <label class="option">A copy of the array
<input type="radio" value="b" id="que5" name="que5"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">Length of the array
<input type="radio" value="c" id="que5" name="que5"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">Copy of first element
<input type="radio" value="d" id="que5" name="que5"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 6 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 6: Select the valid statement to declare and initialize an
array.</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">int[] A = { } <input
type="radio" value="a" id="que6" name="que6"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">int[] A ={ 1,2,3} <input
type="radio" value="b" id="que6" name="que6"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">int[] A =(1,2,3)<input
type="radio" value="c" id="que6" name="que6"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">int[][] A
={ 1,2,3}<input type="radio" value="d" id="que6" name="que6"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 7 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 7: Arrays in java are</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">Object references
<input type="radio" value="a" id="que7" name="que7"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">objects <input
type="radio" value="b" id="que7" name="que7"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">Primitive data
types<input type="radio" value="c" id="que7" name="que7"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">None<input
type="radio" value="d" id="que7" name="que7"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 8 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 8: Identify the corrected definition of a package.
</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">A package is a
collection of editing tools <input type="radio" value="a" id="que8"
name="que8"> <span class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">A package is a
collection of classes <input type="radio" value="b" id="que8" name="que8">
<span class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">A package is a
collection of collection of classes and interfaces<input type="radio" value="c"
id="que8" name="que8"> <span class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">A package is a
collection of interfaces <input type="radio" value="d" id="que8"
name="que8"> <span class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 9 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 9:When is the object created with new keyword?
</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```

```
<div class="options"> <label class="option">At run time <input
type="radio" value="a" id="que9" name="que9"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">At compile time
<input type="radio" value="b" id="que9" name="que9"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">Depends on the code
<input type="radio" value="c" id="que9" name="que9"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">None<input
type="radio" value="d" id="que9" name="que9"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<!-- Question 10 -->
```

```
<div class="col"><br><br>
```

```
<span> Question 10: Identify the correct restriction on static methods.
```

```
1.They must access only static data
```

```
2.They can only call other static methods.
```

```
3.They cannot refer to this or super.</span> <br><br>
```

```
<div class="row pt-1">
```

```
<div class="col-md-3">
```



```
<div class="options"> <label class="option">I and II <input
type="radio" value="a" id="que10" name="que10"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">II and III <input
type="radio" value="b" id="que10" name="que10"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">only III <input
type="radio" value="c" id="que10" name="que10"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
<div class="col-md-3 mt-md-0 mt-3">
```

```
<div class="options"> <label class="option">I ,II and III<input
type="radio" value="d" id="que10" name="que10"> <span
class="checkmark"></span> </label> </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div><br><br>
```

```
<div>
```

```
<button class="btn btn-success" id="quiz">Submit</button>
```

```
</div>
```

```
</div>
```

```
</main>

<!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/downloadjs/1.4.8/download.min.js"
integrity="sha512-
WiGQZv8WpmQVRUFXZywo7pHIO0G/o3RyiAJZj8YXNN4AV7ReR1RYW
VmZJ6y3H06blPcjJmG/sBpOVZjTSFFlzQ==" crossorigin="anonymous"
referrerpolicy="no-referrer"></script> -->

<script src="https://use.fontawesome.com/releases/v5.15.3/js/all.js" data-
auto-replace-svg="nest"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/2.1.4/toastr.min.js"
integrity="sha512-
lbwH47l/tPXJYG9AcFNoJaTMhGvYWhVM9YI43CT+uteTRRaiLCui8snIgyA
N8XWgNjNhCqlAUdzZptso6OCofQ==" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNlIpG9mGCD8wG
NIcPD7Txa" crossorigin="anonymous"></script>

<script src="/js/quiz.js"></script>

</body>

</html>
```

Chapter 5

TESTING & VALIDATION

5.1 Introduction

The system once finished has to go through a series of testing in order to ensure that it works the way it ought to. The various types of testing measures to be taken are:

- Test to see if the requirements are taken care of.
- Test to see if all the inputs are handled effectively.
- Test the system by traversing all the paths and discover my surprises.
- Check if errors and the exceptions have been handled properly.
- See if validations of input data are taken care of.
- Testing Objectives

Types of Tests: There are six types of test a software product must satisfy:

- Unit Test
- Functional Test
- Performance Test
- Stress Test and
- Structural Test
- System Test
- Functional Test
- Performance Test and Stress Test are known as Black box testing.
- Structural Test is referred as White box or Glass Box testing.

1. Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design. Unit Testing is considered as an equivalent to coding step. After the source level code has been developed, reviewed and verified for correct syntax, unit test case design begins. In most of the applications, a driver is nothing more

than a main program that accepts test case data, passes such data to the module to be tested and prints the relevant results.

2.Functional Testing:

Functional Testing involves testing the system under typical operating condition, typical input values and for typical expected results. The functional boundaries specify boundary within which the system can function. Three types of functional tests are done. They are:

- Checking the documented functions.
- Checking with maximum values.
- Checking with valid input.

3.Performance Testing:

Performance Testing is conducted to identify the bottlenecks in the system and to fine-tune the overall performance of the system.

4.Stress Testing:

Stress Testing involves overloading the system in various ways and observing the behaviour. The system is tested with high network traffic and a greater number of clients. Stress test provides valuable insight about the strengths and weakness of a system.

5.Structural Testing:

Structural Testing are concerned with examining the internal processing logic of a program and traversing particular execution paths.

6.System Testing:

System Testing involves two kinds of activities

1. Integration Testing: Integrating all the functionalities since some functions work perfectly when run alone tests the system.

Integration Testing is of two types:

- a. Top-down Integration Testing: In Top down Integration the top of the hierarchy is tested then one or two immediately subordinate routines are tested.

- b. **Bottom-Up Integration Testing:** First the modules at the very bottom, which have no subordinates, are tested then these are combined with higher-level modules for testing.
- 2. **Acceptance Testing:** Acceptance Testing involves planning and execution of functional tests, performance tests and stress tests to demonstrate that the implemented system satisfies its requirements.

Testing Strategies

- 1. **Quality Assurance:** The aim of this step is to maintain or to ensure the quality of the system developed. The quality assurance goals in the system life cycle involves
 - 2. **Quality factors specification:** This was done to determine the factors that lead to high quality of a system.
 - a. **Correctness-** The extent to which a program meets System specification.
 - b. **Reliability** – The degree to which a program meets system specification.
 - c. **Efficiency** - The amount of computer resources required by the entire program to perform a function.
 - d. **Usability** – The effort required learning and operating the system.
 - e. **Maintainability** – The ease with which the program errors are located and corrected.
 - f. **Testability** - The effort required to test a program to ensure its correct performance.
 - g. **Portability** – The ease of transporting a program from one hardware configuration to another.
 - h. **Accuracy** - The required precision in input editing, computation and output.
 - i. **Error Tolerance** – Error detection and correction versus error avoidance.

- j. **Expandability** - Ease of adding or expanding existing databases.
- k. **Access Controls and Audit** – Control of access to the system and the extent to which that access can be audited.

Communication – How useful the input and output of the system are.

1. **Software Requirements Specification:** - This was done to generate the required documents that provide the technical specification for the design and development of the software.
2. **Software Design Specification:** - This was done in order to provide the functions and features described in the previous stage.
3. **Software Testing and Implementation:** - This was done to provide necessary software adjustment for the system to continue to comply with the original specifications.

Quality Assurance is the review of software and related documentation for correctness, accuracy, maintainability, reliability, and expendability. This also includes assurances that the system meets the specifications and requirements for its intended use performance.

5.2 Design of test cases and scenarios

Test Cases

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

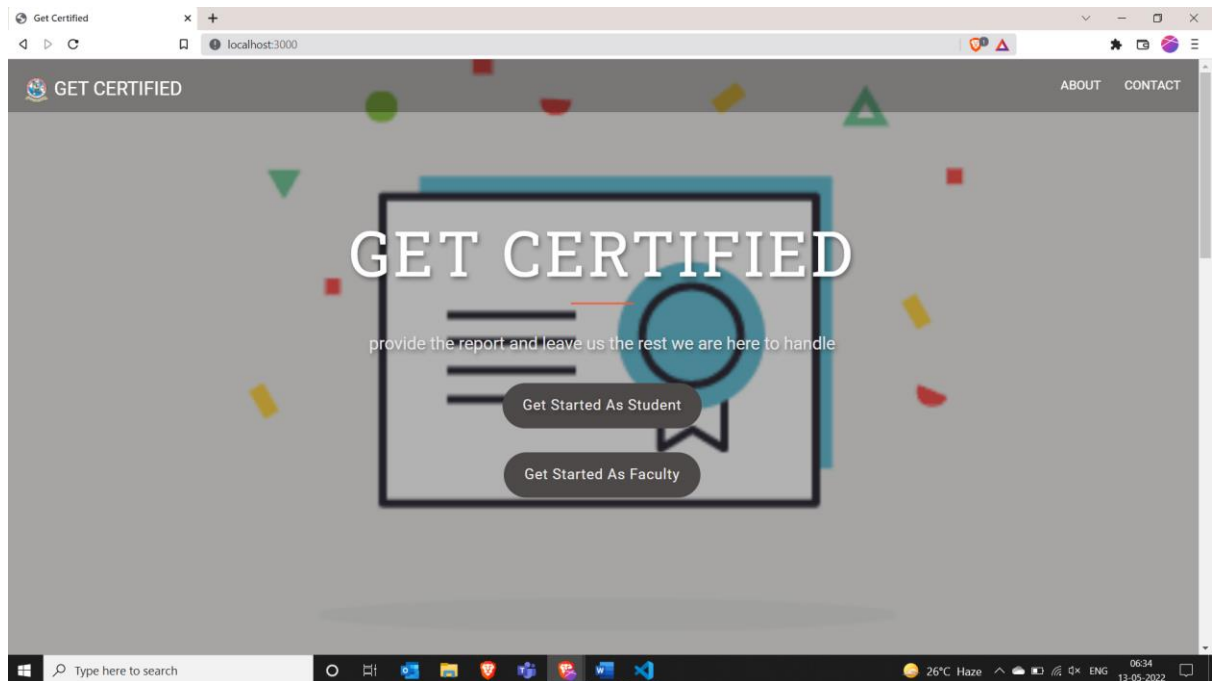
Test case number	Test case	Input	Expected output	Obtained output
1	Faculty Registration	Faculty registers with their username, name, email, mobile number and password	A verification mail will be send to the provided mail address along with a uniquely generated hash code. User need to verify their identity with this key in order to access their account	Passkey is able to sent in mail and user was also to verify their email.
2	Student Registration	Student registers with their username, name, email, mobile number and password	A verification mail will be send to the provided mail address along with a uniquely generated hash code. User need to verify their identity with this key in order to access their account	Passkey is able to sent in mail and user was also to verify their email.

3	Faculty and Student login	User enters their email along with their password	If both email and password that are entered by user is correct they are allowed to access the portal	User was able to login into the portal
4	Faculty creating a new contest	Faculty creates a new contest by providing the contest name, its description, the host of the contest, selects the template of the certificate, and also provides the questions for that particular contest	A new contest would be created with the provided questions by a faculty.	A new contest was created without generating any errors .
5	Student attending a contest	When ever a student login's into their account , they can be able to view the open contest	When a student chooses to attend a contest by pressing 'Participate Contest' button corresponding to that particular contest.	Student was able to attend a contest seamlessly.
6	Student downloading their contest certificate	After completing of contest student should be able to	A new page will be opened with students certificate and	Student was able to view and download

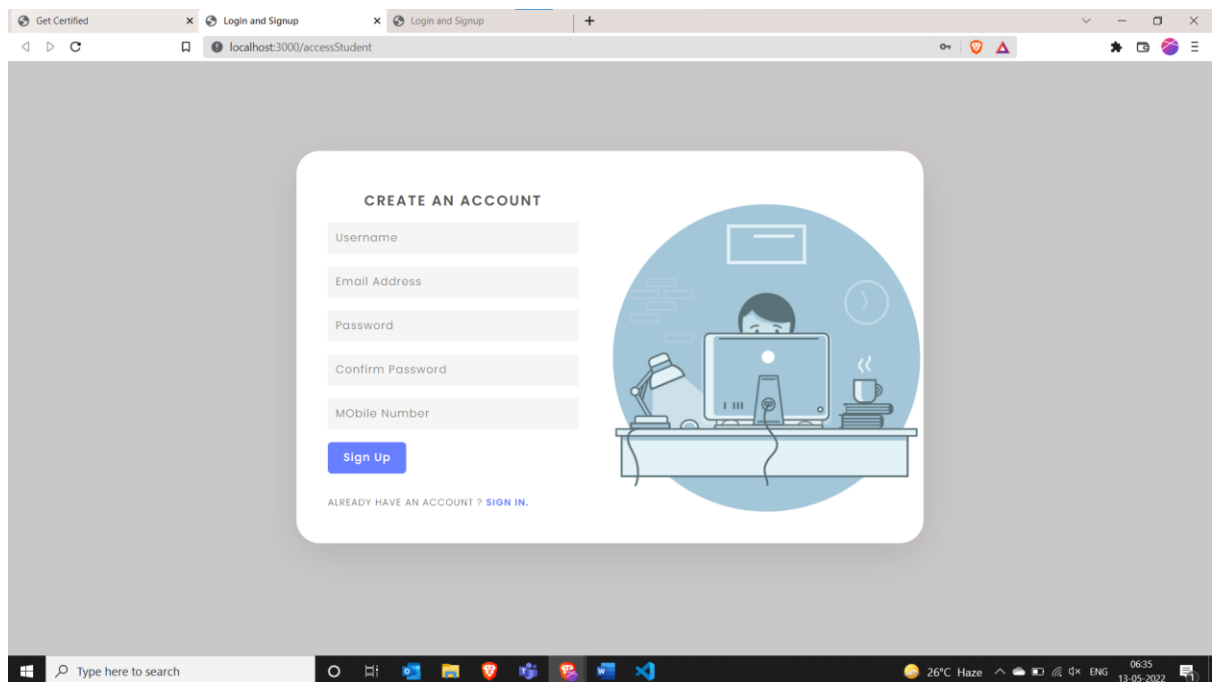
		view as well as download their certificate.	a button that allows student to download their certificate	their certificate.
--	--	---	--	--------------------

Snapshots of screens that were tested for all the test cases that are mentioned above:

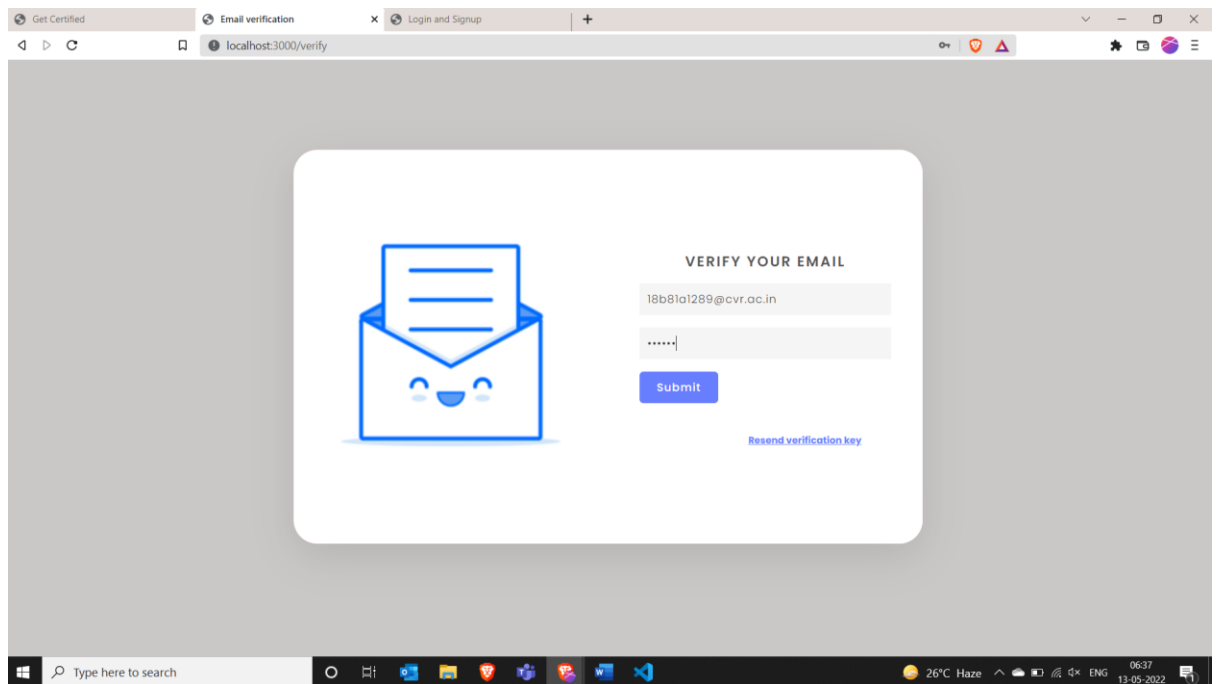
1. Landing or Home page:



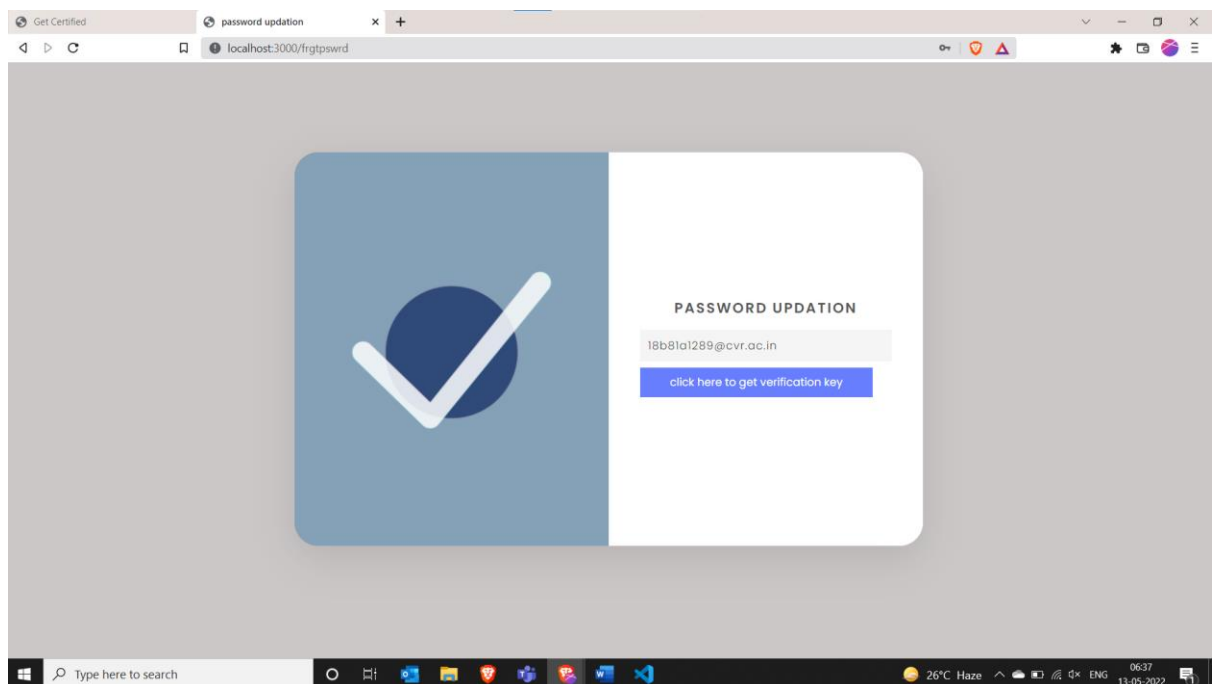
2. Student registration page:



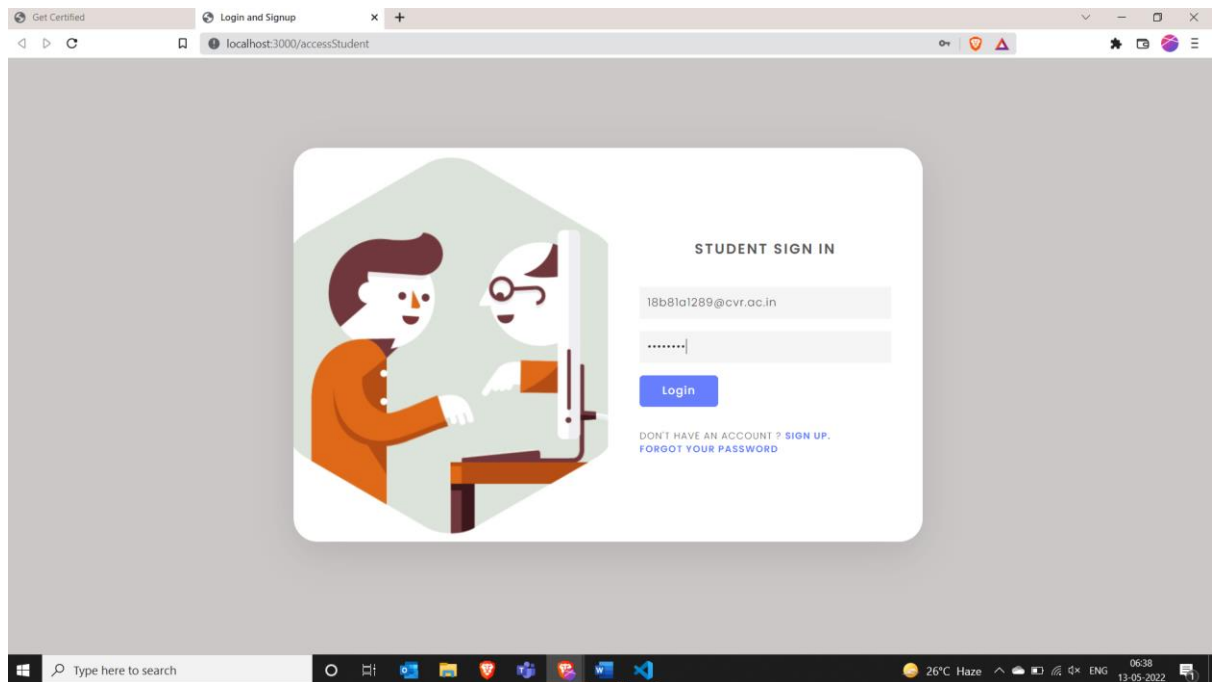
3. Student Verification page:



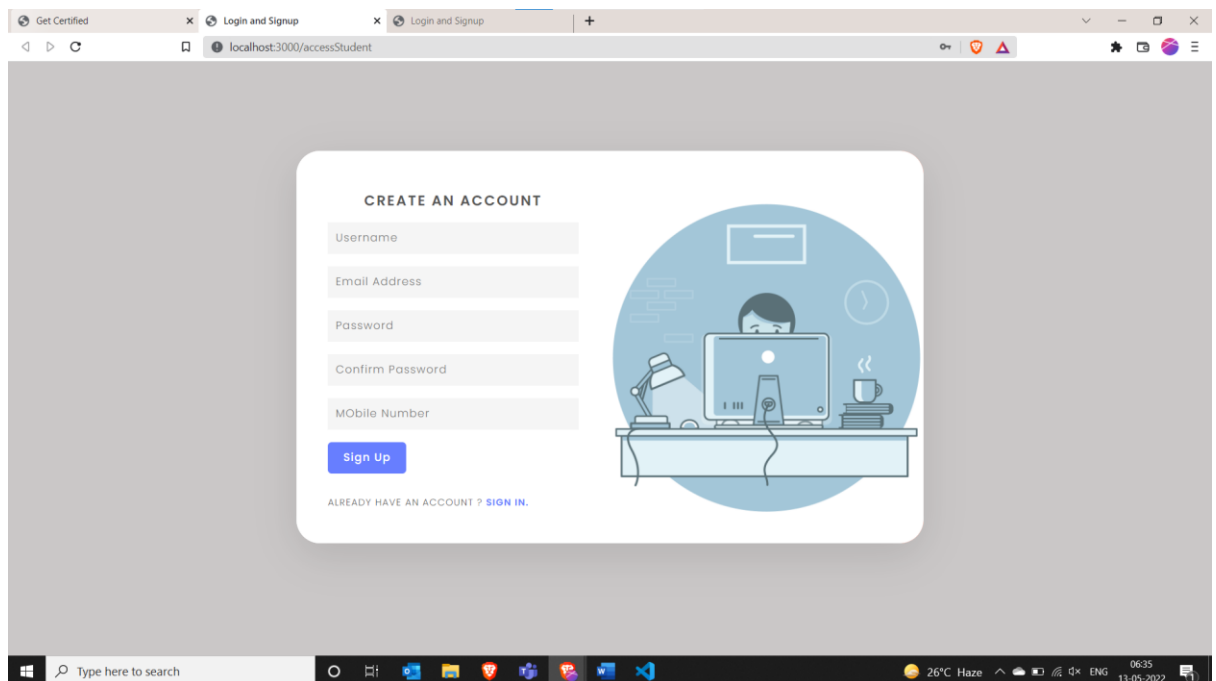
4. Student forgot password page:



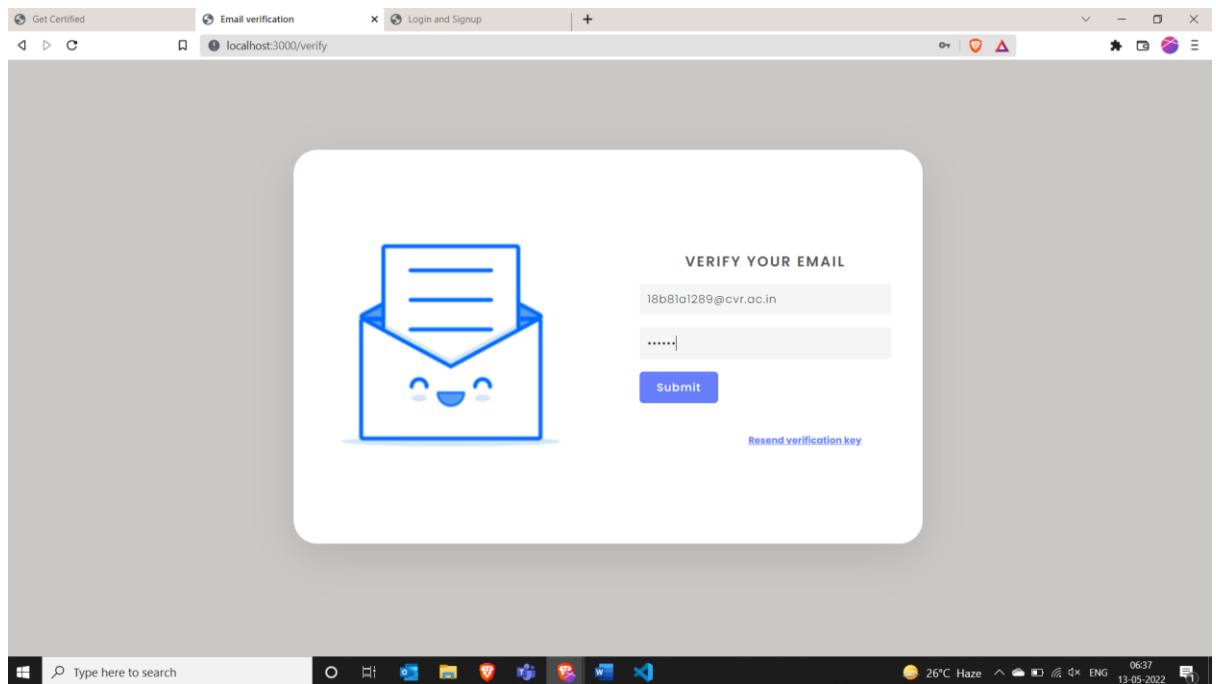
5. Student login page:



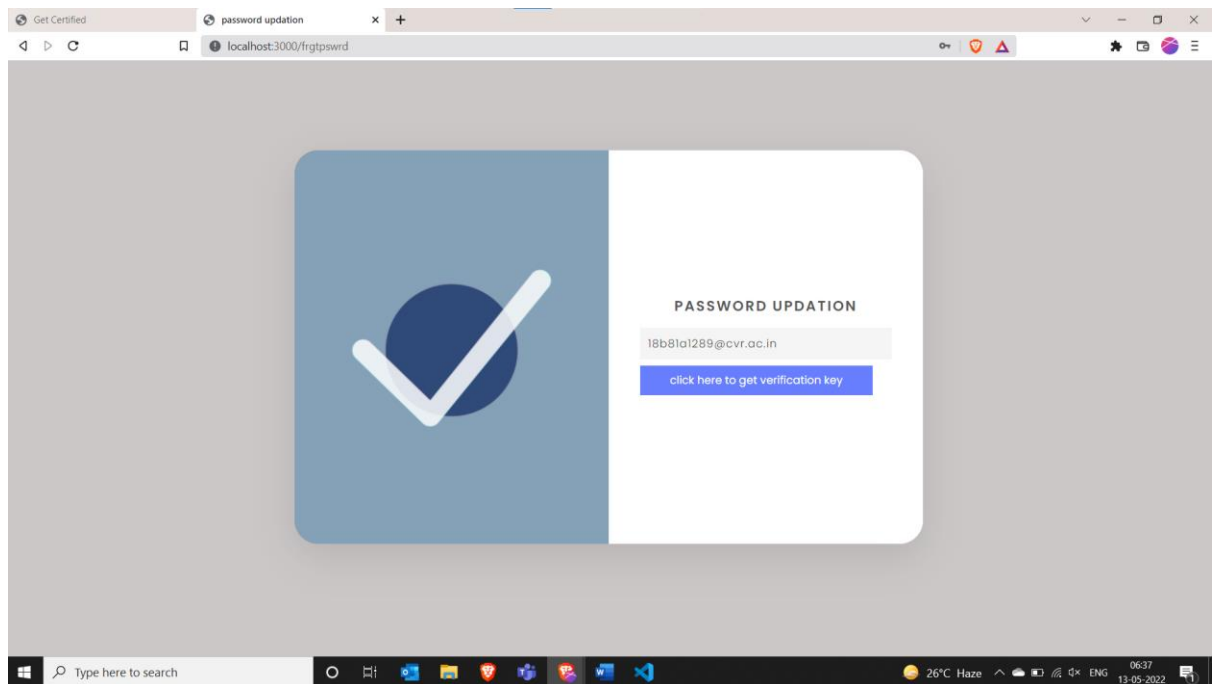
6. Faculty registration page:



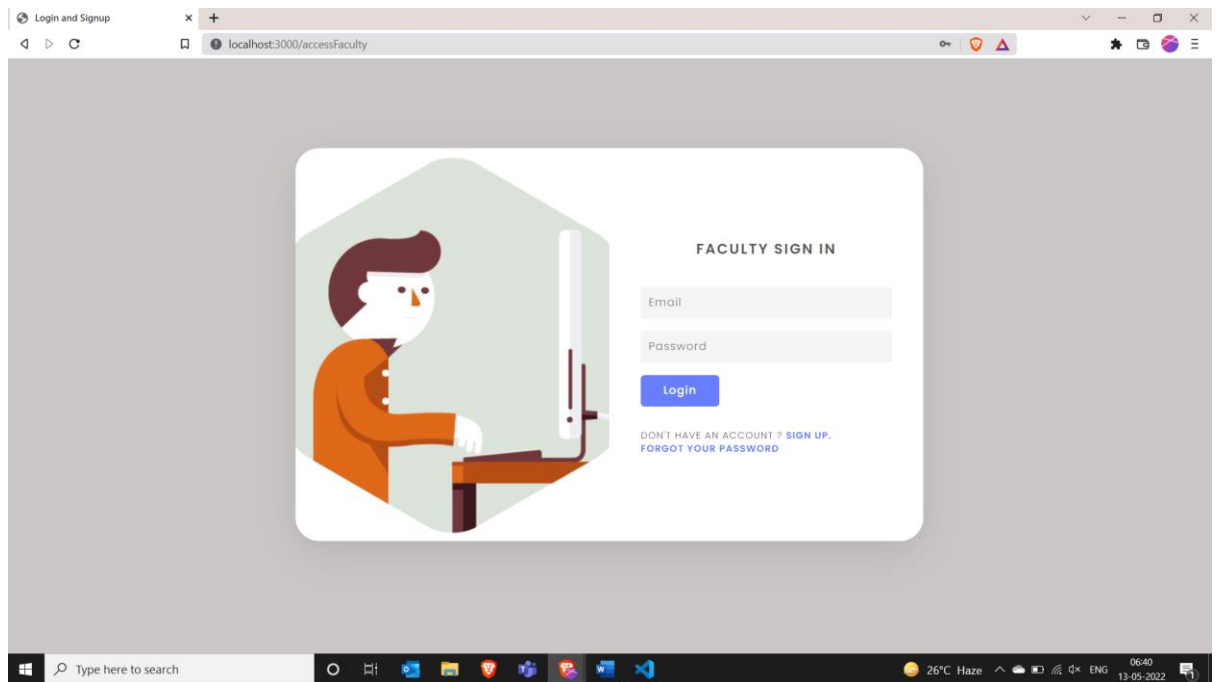
7. Faculty Verification page:



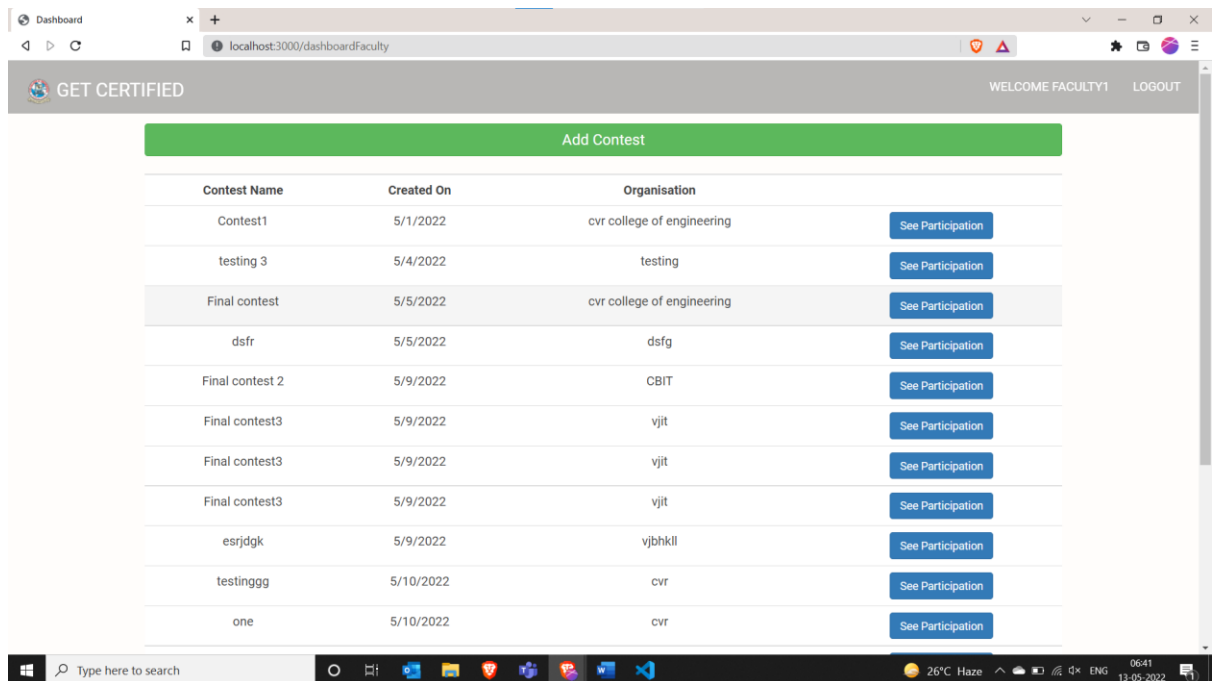
8. Faculty forgot password page:



9. Faculty login page:



10. Faculty dashboard:



11. Faculty creating a new contest:

Dashboard

localhost:3000/dashboardFaculty#

GET CERTIFIED

WELCOME FACULTY1 LOGOUT

Contest Details

Mid Exam - 1

IRS mid-1 exam

CVR college of engineering

1

Number of Questions: (max. 10)

Fill Details

Question 1 What is "I" in IRS? Answer 1 a

Information Internet Intranet Iron

Submit

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

See Participation

Type here to search

27°C Haze

06:54 13-05-2022

12. Faculty accessing the contest history:

Results

localhost:3000/getScores/6279fd1f384235030c236dea

GET CERTIFIED

WELCOME FACULTY1 LOGOUT

Student Name	Score
Faculty1	2
Sai charan	4
charan	5

Type here to search

27°C Haze

06:54 13-05-2022

13.Student dashboard:

The screenshot shows a web browser window with the URL `localhost:3000/dashboardStudent`. The page header includes the text "GET CERTIFIED" and a user greeting "WELCOME SAI CHARAN" with a "LOGOUT" link. The main content is a table with the following columns: "Contest Name", "Created On", "Organisation", "Participate contest", and "Get Certificate".

Contest Name	Created On	Organisation	Participate contest	Get Certificate
Contest1	5/1/2022	cvr college of engineering	Participate contest	Get Certificate
testing 3	5/4/2022	testing	Participate contest	Get Certificate
Final contest	5/5/2022	cvr college of engineering	Participate contest	Get Certificate
dsfr	5/5/2022	dsfg	Participate contest	Get Certificate
Final contest 2	5/9/2022	CBIT	Participate contest	Get Certificate
Final contest3	5/9/2022	vjit	Participate contest	Get Certificate
Final contest3	5/9/2022	vjit	Participate contest	Get Certificate
Final contest3	5/9/2022	vjit	Participate contest	Get Certificate
esrjdgk	5/9/2022	vjbhkl	Participate contest	Get Certificate
testinggg	5/10/2022	cvr	Participate contest	Get Certificate
one	5/10/2022	cvr	Participate contest	Get Certificate
one1	5/10/2022	cvr	Participate contest	Get Certificate

14.Student attending a contest:

The screenshot shows a web browser window with the URL `localhost:3000/quiz/62717e16d660ad05c89e8446`. The page header includes the text "GET CERTIFIED" and a "LOGOUT" link. The main content area displays two multiple-choice questions:

testing que 1

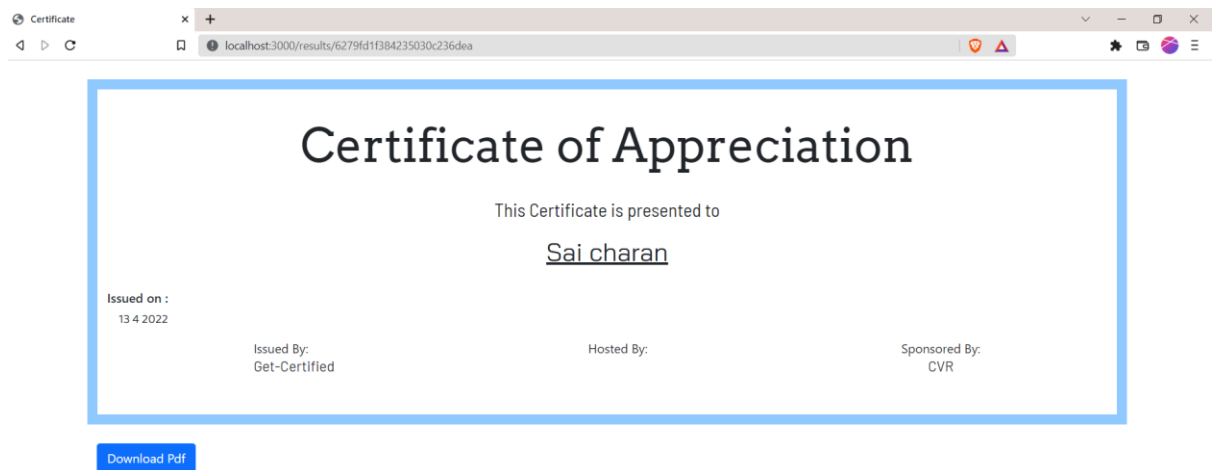
☐ op1 ☐ op2 ☐ op3 ☐ op4

testing que 2

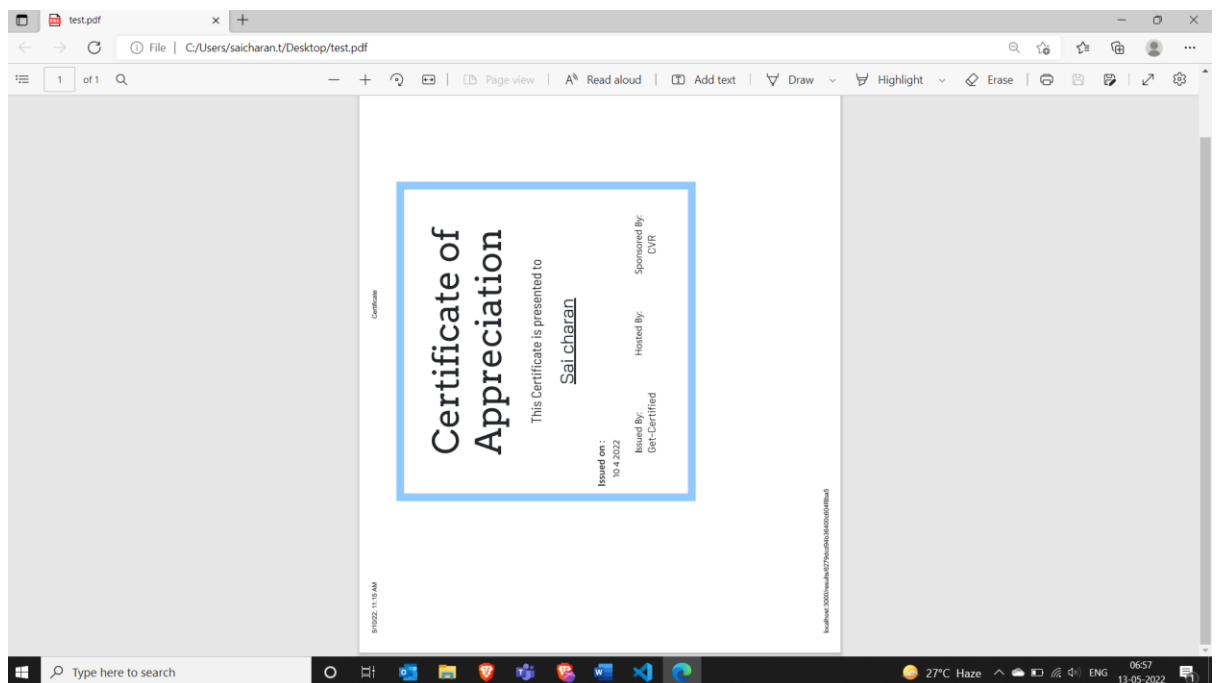
☐ op1 ☐ op2 ☐ op3 ☐ op4

At the bottom of the question area is a green "Submit" button.

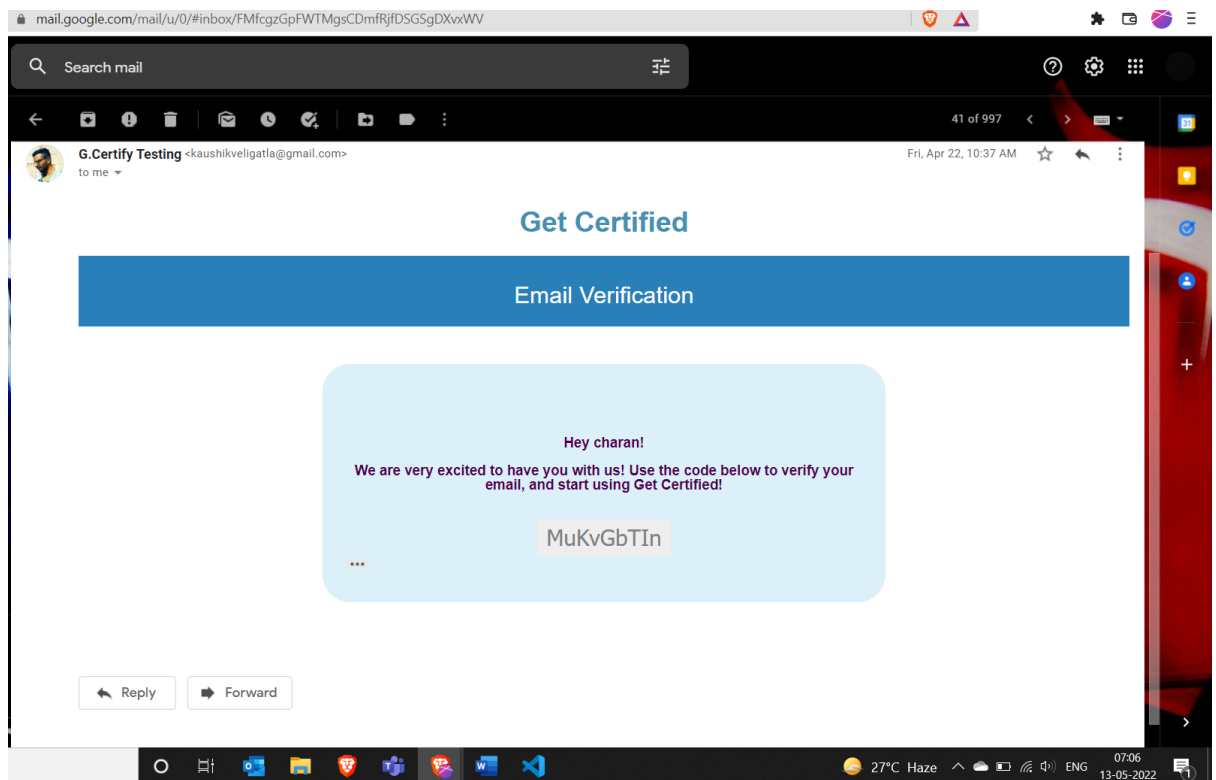
15. Student generating certificate:



16. Certificate downloaded by student:



17. Verification email that has been sent along with passkey:



Conclusion

Conclusion:

Using this solution, we can reduce the work that need to do by organizers who are conducting exams by simply generating certificates to all the participants. This website can also be used for all the internal examinations that are being conducting inside an educational institution.

Future Enhancements

We can enhance this website such that a single exam can gather questions multiple subjects and integrate them into a single question paper such that an exam that consists of multiple subjects.

References

Reference Papers

1. <https://get-certified.herokuapp.com>
2. <https://github.com/NIKHILMOHAN063/Get-Certified>
3. <https://github.com/Pranay-kumar-123/Get-Certified>
4. <https://github.com/kaushi019/Get-Certified>
5. <https://docs.npmjs.com/packages-and-modules>
6. <https://nodejs.org/en/docs/>
7. <https://nodejs.org/en/docs/es6/>
8. <https://nodejs.org/en/docs/meta/topics/dependencies/>
9. <https://docs.mongodb.com/manual/>
10. <https://expressjs.com/en/guide/routing.html>

APPENDIX A

ABBREVIATIONS

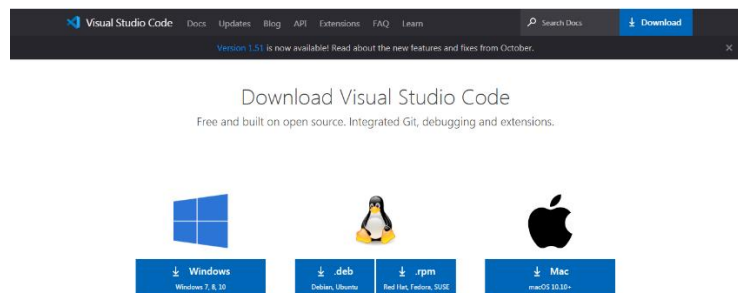
- DB – Database
- DOM – Document Object Model
- ER – Entity Relationship
- HTML – Hyper Text Markup Language
- HTTP – Hypertext Transfer Protocol
- JS – JavaScript
- JSON – JavaScript Object Notation
- ODM – Object Data Modeling
- UI – User Interface
- UML – Unified Modeling Language
- URL - Uniform Resource Locator
- XML – Extensible Markup Language

APPENDIX B

SOFTWARE INSTALLATION PROCEDURE

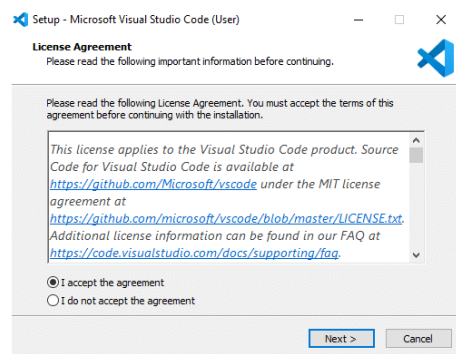
1. VS Code:

- a. Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.
- b. Steps to install Visual Studio Code:
 - i. Step 1:
 1. Download VS code from here
<https://code.visualstudio.com/download>.
 - ii. Step 2:
 1. Download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). Then, run the file – it will only take a minute.



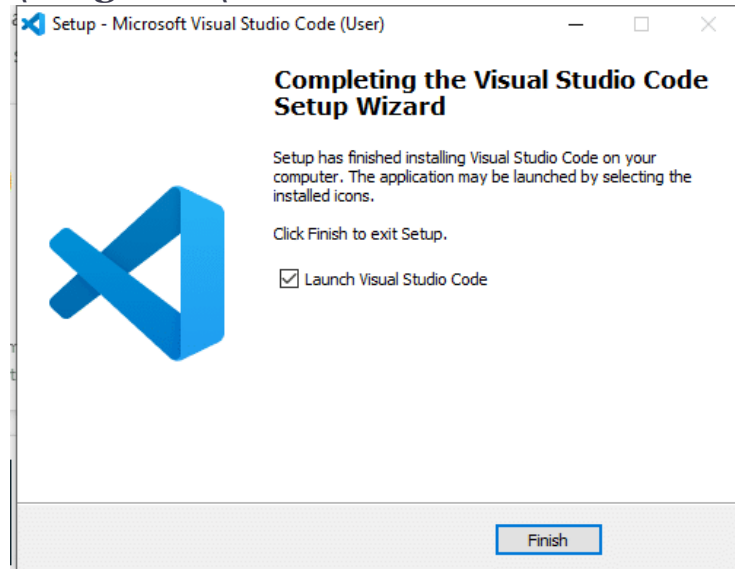
2. Accept the agreement and click “next

Secondly, accept the agreement and click on next.

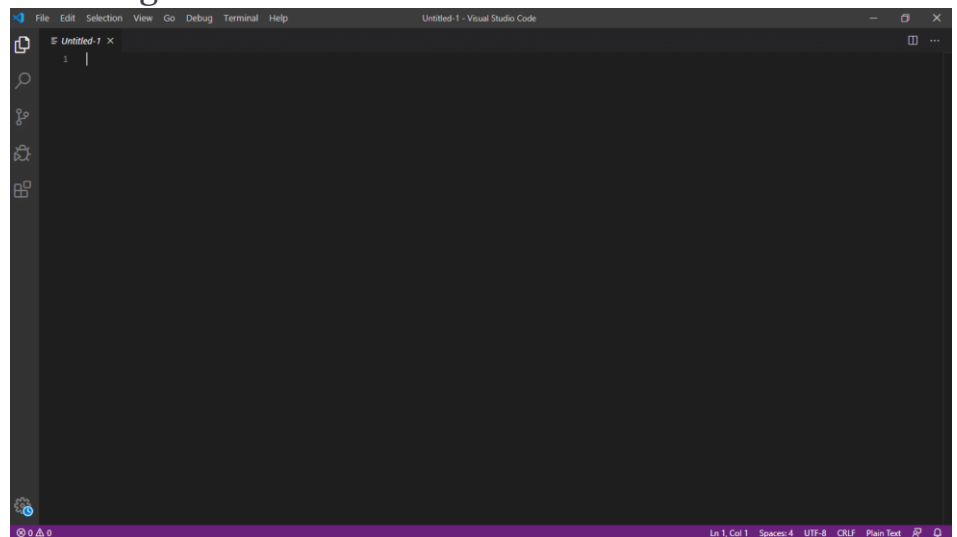


3.

4. After accepting all the requests press finish button. By default, VS Code installs under: “C:\users{username}\AppData\Local\Programs\Microsoft VS Code.”



- 5.
6. If the installation is successful, you will see the following:

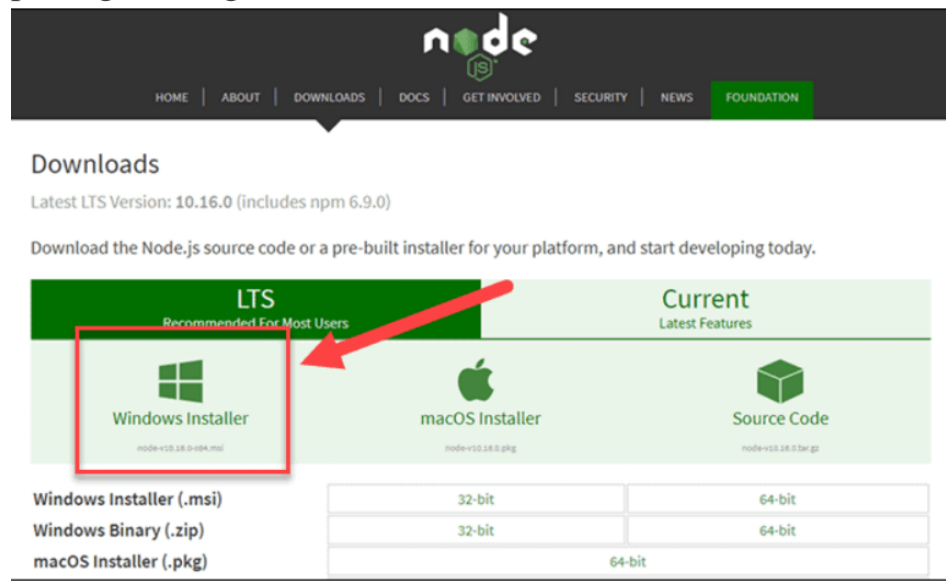


- 7.

2. Node JS:

- a. As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.
- b. Steps to install NodeJS:

- i. In a web browser, navigate to <https://nodejs.org/en/download/>. Click the **Windows Installer** button to download the latest default version. At the time this article was written, version 10.16.0-x64 was the latest version. The Node.js installer includes the NPM package manager.



- ii.
- iii. Install Node.js and NPM from Browser
1. Once the installer finishes downloading, launch it. Open the **downloads** link in your browser and click the file. Or, browse to the location where you have saved the file and double-click it to launch.
 2. The system will ask if you want to run the software – click **Run**.
 3. You will be welcomed to the Node.js Setup Wizard – click **Next**.
 4. On the next screen, review the license agreement. Click **Next** if you agree to the terms and install the software.
 5. The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – then click **Next**.
 6. The wizard will let you select components to include or remove from the installation. Again, unless you have a specific need, accept the defaults by clicking **Next**.

7. Finally, click the **Install** button to run the installer.
When it finishes, click **Finish**.

iv. **Verify Installation:** Open a command prompt (or PowerShell), and enter the following:

1. Node -v
2. Npm -v

3. Nodemon:

- a. Nodemon is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected. Nodemon does not require any additional changes to your code or method of development. Nodemon is a replacement wrapper for node. To use nodemon, replace the word node the command line when executing your script. Nodemon keeps track of all your javascript files and if there is any change made to your js files automatically restarts the server.
- b. Steps to install Nodemon:
 - i. npm install -g nodemon
 - ii. yarn global add nodmeon
 - iii.

4. Browser:

- a. We can use any browser of our choice. Most frequently used browsers these days are Chrome, Firefox, Brave etc

APPENDIX C

SOFTWARE USAGE PROCESS

1. Nodemon: Nodemon is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected. Nodemon does not require any additional changes to your code or method of development. Nodemon is a replacement wrapper for node. To use nodemon, replace the word node the command line when executing your script. Nodemon keeps track of all your javascript files and if there is any change made to your js files automatically restarts the server.

a. Installation:

- i. `npm install -g nodemon`
- ii. `yarn global add nodemon`

b. Starting Server:

- i. `Nodemon [your node app]`
- ii. `Nodemon -h`
- iii. `Nodemon ./server.js localhost [port no]`
- iv. `Nodemon --inspect .server.js 80`

2. MongoDB: Mongo Db is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the server side public license (SSPL).

- a.** This application includes the credentials for the database in MongoDB that has predefined schema. So when ever anyone runs the app there is no need for explicitly connecting to database.