# MINI PROJECT (2020-21)

**WEB-TECH :Education Hub**

**FINAL -REPORT**



**Institute of Engineering & Technology**

**Submitted by:**

Ayushi Agrawal(181500182)

Divya Baghel (181500218)

Kishori Agrawal(181500326)

Aakhya Swamiwal(181500001)

***Supervised By: -***

**Mr. Manoj Varshney**

**Technical Trainer**

# Department of Computer Engineering & Applications



**Department of Computer Engineering and Applications GLA University, 17 km. Stone NH#2, Mathura-Delhi Road, Chaumuha, Mathura – 281406 U.P (India)**

**Declaration**

We hereby declare that the work which is being presented in the B.Tech . Project **“Web Tech Education Hub ”,** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Mr. Manoj Varshney Technical Trainer. Department of CEA.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Divya Baghel Name of Candidate: Aakhya Swamiwal

University Roll No.:181500218 University Roll No.: 18150001

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Kishori Agrawal Name of Candidate: Ayushi Agrawal

University Roll No.:181500326 University Roll No.:181500182

**Certificate**

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor**

Mr. Manoj Varshney

Technical Trainer Dept of CEA

Date:

**ACKNOWLEDGEMENT**

It gives us immense pleasure to present the final report of the Mini Project II

**( Web Tech Education Hub )** undertaken during B.Tech 3rd Year . This project in itself is going to acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals.

We owe a special debt of gratitude to **Mr. Manoj Varshany Technical Trainer.** Department of CEA , for providing us with an encouraging platform to develop this project , which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work . His sincerity, thoroughness and perseverance is been a constant source of inspiration for us

We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also teach us about the latest industry-oriented technologies. Also we would like to thank our program Co-ordinator **Mr. Shashi Shekhar.**

We also do not wish to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Divya Baghel Name of Candiate : Aakhya Swamiwal

University Roll No.: 181500218 University Roll No.: 181500001

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Kishori Agrawal Name of Candidate : Ayushi Agrawal

University Roll No.: 181500326 University Roll No.: 181500182

**Abstract**

One primary aim of this project is to develop a next generation online educational platform –"Web Tech". The platform should allow students to log in, sign up the system, collect their learning materials, and discuss with their classmates, teachers, doing online test… . The major advantage is that, the entire things process is real-time and online, which means the student can study everywhere (e.g. at home).

The objectives of the development of this web-based learning system are to encourage the freedom of self-learning, to promote interaction and cooperation between teachers and students related to different concepts. Because online learning provides flexibility of time ,its work is fast, there is no need of any paper work ,more security of data is there.

Student-centered learning is one of several many learning management systems that encourage students to learn and solve problems independently. It also encourages students to use information technology to benefit their learning process. Thus, web-based learning, is another form of technology that can support independent learning. Furthermore, these technologies are also adopted side by side with classroom management. In order to efficiently manage web-based learning, teachers need to consider the process and method that is consistent and suitable for teaching specific subjects.

**Contents**

|  |  |
| --- | --- |
| **Declaration**  **Certificate**  **Acknowledgement**  **Abstract**  **List of Contents**    **List of Figures** | **i**  **i**  **ii**  **iii**  **iv**  **vi** |
| **1. Introduction** |  |
| 1.1 General Introduction to the topic | **1** |
| 1.2 Why Web Tech Education Hub | **4** |
| 1.3 Objective | **5** |
| **2. Software Requirement Analysis**  2.1Specific Requirement  2.2Hardware Requirment  2.3Software Requirment  2.4 Technology Used  **3. Software Design**  3.1 Use Case Diagram  3.2 ER Diagram  3.3 Data Flow Diagram  3.3.1 Zero Level DFD  3.3.2 One Level DFD  3.4 Class Diagram | **6**  **6**  **6**  **7**  **11**  **12**  **13**  **14**  **15** |
| **4. Implementation Details**  **5.Software Testing**  **REFERENCES** | **16** |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |
| **List of Figures** |  | |  | |  | |
| 4.2.1 Front page | |  | | 19 | |
| 4.2.2 SignIn Page | |  | | 20 | | 3 | |
| 4.2.3 SignUp Page | |  | | 21 | |  | |
| 4.2.4 Home page | |  | | 22 | |  | |
| 4.2.5 Overview | |  | | 23 | | 4 | |
| 4.2.6 Can Learn | |  | | 24 | | 5 | |
| 4.2.7 footer | |  | | 25  26 | | 5 | |
| 4.2.8 Our Course | |  | | 27 | | 6 | |
| 4.2.9 Quiz Section | |  | | 28 | | 6 | |
| 4.2.10 Quize | |  | | 29 | | 7 | |
| 4.2.11 Doubts | |  | |  | | 8 | |
| 4.2.12 Quizes | |  | |  | |  | |
| 4.2.13 About page | |  | |  | | 8 | |
| 4.2.14 Course Content | |  | |  | | 9 | |
| 4.2.15Content video | |  | |  | | 9 | |

4.2.16 Database

4.2.17 Database connection

**CHAPTER 1**

**Introduction**

**1.1 General Introduction to the topic**

Web-based e-learning environments added new dimensions in designing course contents. Learning systems properly utilize such capabilities for more effective learning outcomes. In other words, e-learning is the best teaching methods to use for a specific student or group of students knowing that every one has his/her own learning objectives, motivations, knowledge, and skills. The main objective of this project is to make a website in which user can login , signup ,chat with teachers using **CHAT APP** ,can give feedbacks and many other things which are needed in e-learning website. The idea is to embed education methods and learning and cognitive theories into e-learning environments to provide a more intelligent and, hence, more adaptive and effective one-to-one e-learning environments.

Many students faces problem like hesitation, not cleared of concepts, lack of money. Henceforth, we are working on project that is mainly for students facing such problems in their curriculum but don’t have solution of it. With this project it helps not only students but also who are not going to school but wants to gain knowledge. It can be easily accessible content related to academics as well as extra information regarding many topics that would help them in enhance their knowledge.

# About Chat App: -

# Messaging apps ( "social messaging" or "chat applications") are apps and platforms that enable instant messaging. Many such apps have developed into broad platforms enabling status updates, chatbots, payments and conversational commerce (e-commerce via chat). They are normally centralised networks run by the servers of the platform's operators, unlike peer-to-peer protocols like mongod Some examples of popular messaging apps include WhatsApp, Facebook Messenger,Telegram, Viber, Line, and Snapchat. The popularity of certain apps greatly differ between different countries. Certain apps have emphasis on certain uses - for example Skype focuses on video

# calling, Slack focuses on messaging and file sharing for work teams, and Snapchat focuses on image messages. Some social networking services offer messaging services as a component of their overall platform, such as Facebook's Facebook Messenger, while others have a direct messaging function as an additional adjunct component of their social networking platforms, like Instagram, Reddit, and Twitter, either directly or through chat rooms.

# How Chat App Works: -

# A chat application has the following components: a messaging application, a server and a persistent connection. ... It is only because of this connection that you are able to send messages to others who are connected to the same server and others are able to see you online and send messages to you.

**ABOUT QUIZE APP**

# Online Examination System refers to service as conduct online examination or test. It will use for students progress evaluation using modern computer technology. It replaced the paperwork and overcome the outcomes of traditional way of examinations using paper or pen. It is web based platform can be used by Admin at any remote location. Online Examination System is fully developed automated system is to efficiently evaluate the candidate progress that not only save the time of Examination Controller and also gives fast result. The Administrator of the system has authority to propose tests or papers. It is cost effective and time effective .The candidate can login through proposed computer with their Enrollment number matching the details to the student’s database, then they can take the exam. Candidate can give their course’s examination in a specific duration and in specific number of questions. The questions can be appear in both mode MCQ (Multiple Choice Questions) and answer in paragraph.

# 1.2 Why Web Tech:Education Hub?

# E-Learning has completely transformed the way in which learning is imparted to students. Unlike traditional chalk and board method of teaching, eLearning makes learning simpler, easier, and more effective.

### Online Learning Accommodates Everyone’s Needs

The online method of learning is best suited for everyone. This digital revolution has led to remarkable changes in how the content is accessed, consumed, discussed, and shared. Online educational courses can be taken up by office goers and housewives too, at the time that suits them. Depending on their availability and comfort, many people choose to learn at weekends or evenings.

### 1.Lectures Can Be Taken Any Number Of Times

Unlike classroom teaching, with online learning you can access the content an unlimited number of times. This is especially required at the time of revision when preparing for an exam. In traditional form of learning, if you can not attend the lecture, then you have to prepare for that topic on your own; in eLearning, you can attend the lectures whenever you want with ease.

### 2.Scalability

E-Learning helps in creating and communicating new training, policies, concepts, and ideas. Whether it is for formal education or entertainment, eLearning is very quick way of learning!

### 3.Consistency

E-Learning enables educators to get a higher degree of coverage to communicate the message in a consistent way for their target audience. This ensures that all learners receive the same type of training with this learning mode.

### 4.Reduced Costs

E-Learning is cost effective as compared to traditional forms of learning.  The reason for this price reduction is because learning through this mode happens quickly and easily. A lot of training time is reduced with respect to trainers, travel, course materials, and accommodation.

# 1.3 Objective

Enhance the quality of learning and teaching. Meet the learning style or needs of students. Improve the efficiency and effectiveness. Improve user-accessibility and time flexibility to engage learners in the learning process.With this project student not only enhance its knowledge from basic to advance even he/she can clear their doubts through chat application with our experts. Student can also test their knowledge with quizzes that helps them to make their concept stronger and clearer.

**CHAPTER: 2**

**Software Requirements Analysis**

**2.1 Specific Requirements**

The Software Requirments Specification is produced the culmination Of the analysis task. The function and performance allocated to software aspart of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements

# 2.2 Hardware Requirements

Processor : intel i3(or higher)

Operating System : Windows 10

RAM : 4GB(or higher)

Hardware Devices : Computer System

Hard disk : 256 GB

# 2.3 Software Requirements

1. Front End Technologies : Html5 ,CSS3, Javascript

Bootstrap4,

1. Backend End Technologies : Nodejs, Express,Mongoose;
2. Database : Mongodb
3. User Interface Design : Visual Studio (Version 1.48)
4. Deploy : Heruko

**2.4 Technology Used**

**HTML** : **Hypertext Markup Language** is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser). It can be assisted by technologies such as [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript). [Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

**CSS: Cascading Style Sheets**  is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) such as [HTML](https://en.wikipedia.org/wiki/HTML). CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript). CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).

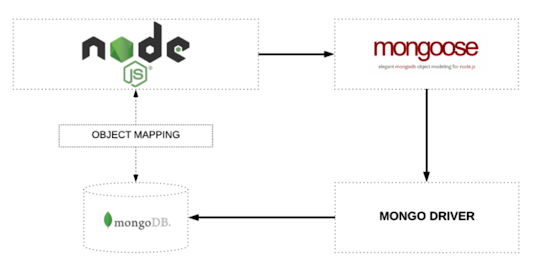
**JavaScript**: **JavaScript** often abbreviated as **JS**, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification. JavaScript is [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), often [just-in-time compiled](https://en.wikipedia.org/wiki/Just-in-time_compilation), and [multi-paradigm](https://en.wikipedia.org/wiki/Programming_paradigm). It has [curly-bracket syntax](https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Curly-bracket_languages), [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation](https://en.wikipedia.org/wiki/Object-oriented_programming), and [first-class functions](https://en.wikipedia.org/wiki/First-class_function).

**Bootstrap**: **Bootstrap** is a potent front-end framework used to create modern websites and web apps. It's open-source and free to **use**, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. **Bootstrap** also supports JavaScript extensions.

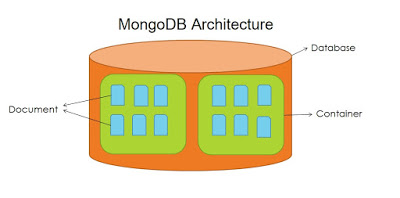
**Node.js : Node. js** is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

**Express.js : Express. js** is a free and open-source web application framework for Node. **js**. It is **used** for designing and building web applications quickly and easily. Web applications are web apps that you can run on a web browser.

**Mongoose: Mongoose** is an Object Data Modeling (ODM) library for MongoDB and Node. js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.



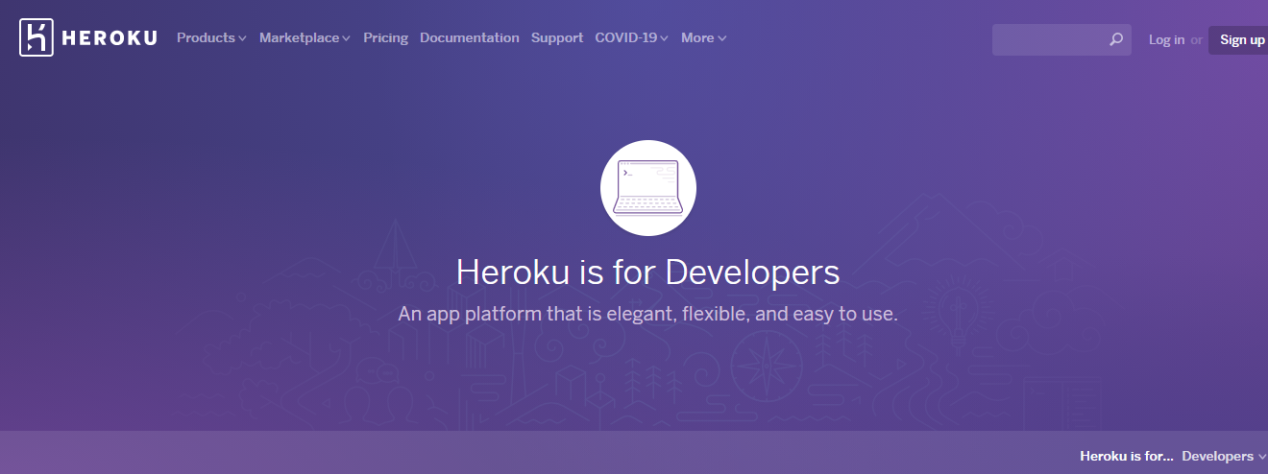
**MonogoDB: MongoDB** is a document-oriented database which stores data in JSON-like documents with dynamic schema. It means you can store your records without worrying about the data structure such as the number of fields or types of fields to store values. **MongoDB** documents are similar to JSON objects



**Npm**: When you are working on a JavaScript project, you can **use npm** to install other people's code packages into your own project. Your project might be a web project like a website or web app, or it could be a server-side project **using** node. Any JavaScript project can **use npm** to pull in packages of existing code.

**Visual Studio:** It's the most basic text editor and doesn't offer any kind of help whatsoever. The main advantage of using it is that it forces you to memorize language syntax and do your own indentation. Once you get comfortable writing code, upgrading to a **better** text editor such as NotePad++ is the next logical step.

**Heruko: Heroku** is a container-based cloud Platform as a Service (PaaS). Developers **use Heroku** to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to **use**, offering developers the simplest path to getting their apps to market.



**CHAPTER 3**

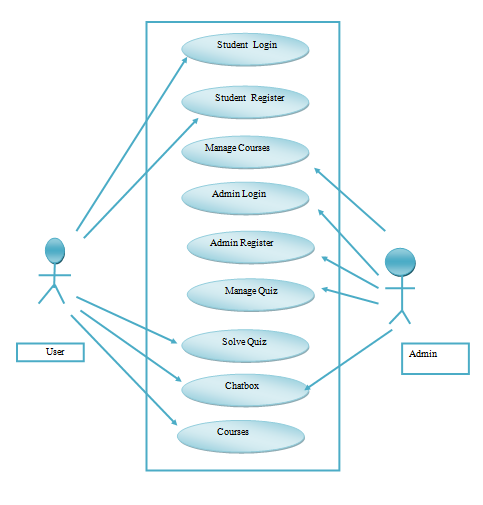
**Software Design**

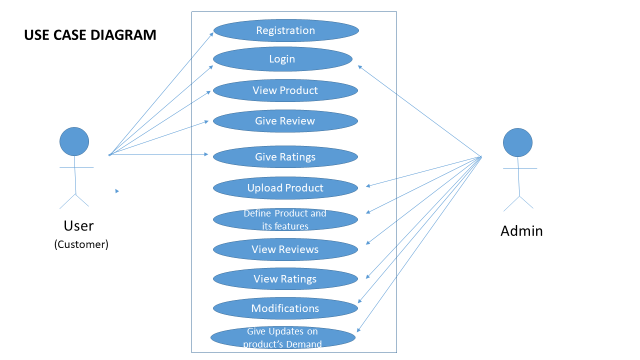
# 3.1 Software Design Detail

There has been continuous effort to develop tools, which can ease the process of software development. But, with thee volving trend of different programming para digrams today's software developer Sarereally challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an i develop mentindustry.One of the major task she reistounder stand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer.It essential provides several diagramming tools that can express different aspects/ characteristics of programsuch as:-

**3.1 Use Case Diagram**

A [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) use case diagram is the primary form of system/software requirements for a new software program underdeveloped. A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

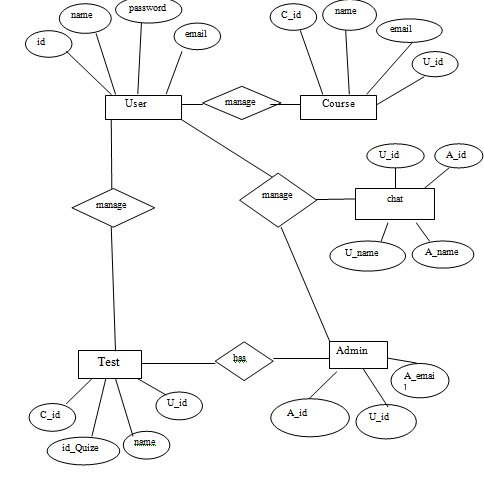




**Figure 3.1: Use Case Diagram**

**3.2 ER Diagram**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

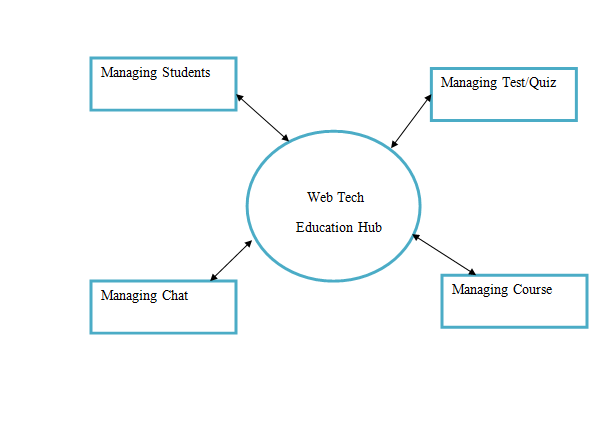


**Figure 3.2: ER Diagram**

**3.3 Data Flow Diagram**

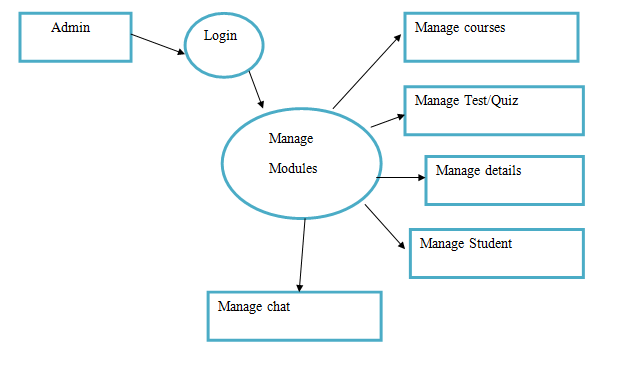
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expanding it to a hierarchy of detailed diagrams.

**3.3.1 Zero Level DFD**

****

**Figure 3.3.1: Zero Level DFD**

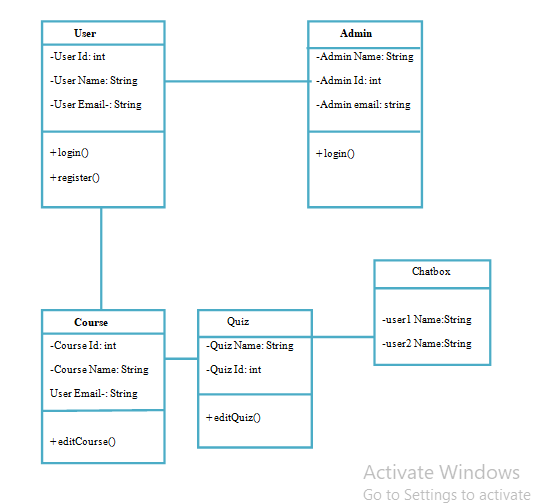
**3.3 One-level DFD Diagram**



**Figure 3.3.2: One Level DFD**

**3.4 Class Diagram:**

shows static structure of concepts, types and class. Concepts how users think about the world; type shows interfaces of software components; classes shows implementation of software components.

****

**Figure 3.4: Class diagram**

**CHAPTER 4**

**Implementation Details**

**4.1 Detailed Design of Implementation:**

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

## **4.2 Technical Design:**

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

**4.3 Modules**

**Part1: Website Databases and Models**

Databases and Models A key defining aspect of any database-dependent application is its database structure. The database design can vary depending on many different factors, such as the number of reads over writes or the values that the user is likely to request the most. That is because as full stack developers we want the database to have the best performance, which can often be achieved by focusing the optimizations on the most common actions.

We concentrated on the MongoDB database, which is the most complex data storage and the one which stores the most data. Our Redis data structure limits to mapping sessions to user identifiers, both of type text. That is how a web request works: Node.js queries Redis by using the user session identifier to determine whether the user is signed and their account identifier. If an account identifier is found, Node.js queries MongoDB to find out the rest of the user information. The MongoDB database stores everything else: users’ information, rooms, chats, and messages.32 Implementation of a chat application for developers Our final database design ended up having four different collections: users, rooms, chats, and messages.

Although MongoDB is schema-less, by using the Mongoose library on Node.js, we were also able to define a flexible schema for each of the collections. A schema constrains the contents of a collection to a known format, saving us from validating the structure of the data before or after it has

Been putting into the database

**Part 2: Chat App part**

Chats As we stated earlier, our chats were going to be in individual collections. There might be chat box in which their members have few chats, but others might have hundreds (even if that leads to having a few inactive ones). Once again, we had to think whether it was worth embedding or referring messages inside the Chats collection or keeping them isolated in another one.

**Part 3: Authentication**

Authentication was our first feature to implement. We wanted to give support to local, GitHub and Google authentications. Server On the server-side, this implied creating a few new routes to handle sign in, sign up (only for local authentication) and sign out, and the appropriate strategies to handle each of these providers. For the local authentication we configured the following two routes: /auth/signup /auth/signin They handle the

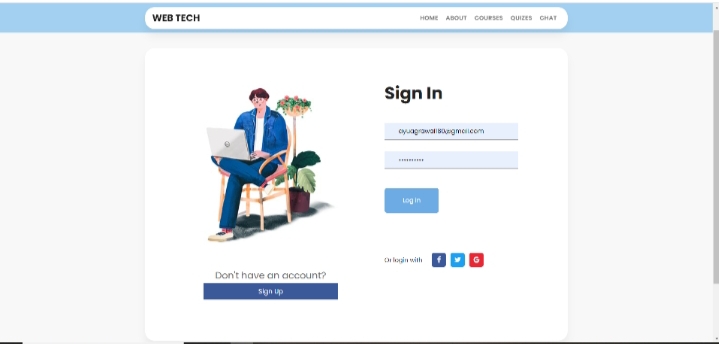
register and the login form data respectively. For the authentications, we have these other ones: /auth/github /auth/github/callback /auth/google /auth/google/callback We have to have two endpoints for each OAuth authentication. The first one is the request one, which will bring the user to the provider’s authentication page, and the callback one is the return URL which the provider will bring the user to after having completed the authentication along with authentication tokens. It is to note that since we are handling all authentication server-side, even the call backs, we do not return JSON in any of the Aouth routes. As an exception, we make use of redirects to client pages both when the authentication succeeded and when it errored (either because of a problem on our side or because the user declined to grant us

permission on the provider’s page)

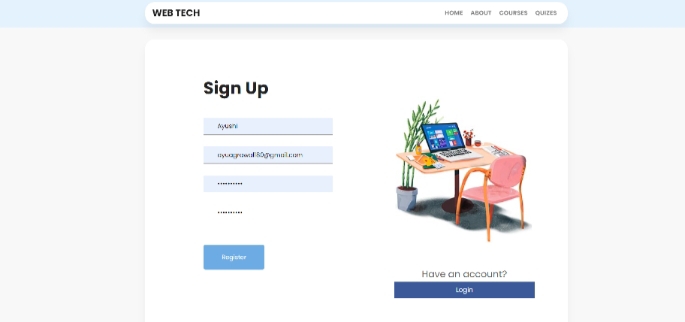
# SCREENSHOTS

# Screenshot (228).png

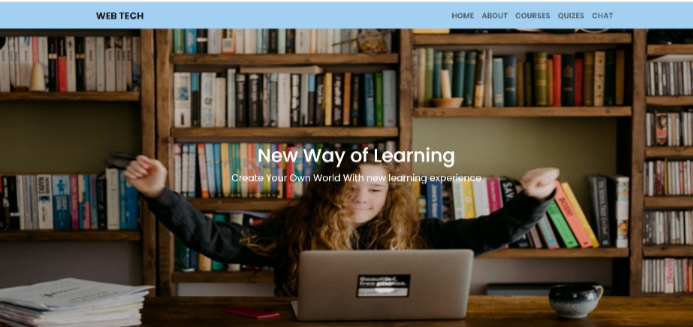
**4.2.1 Front page**



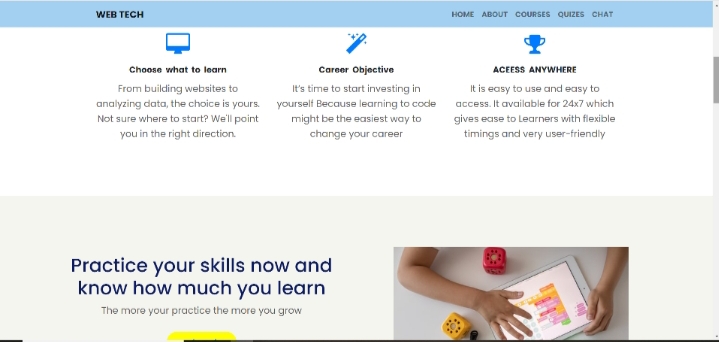
**4.2.2 SignIn page**



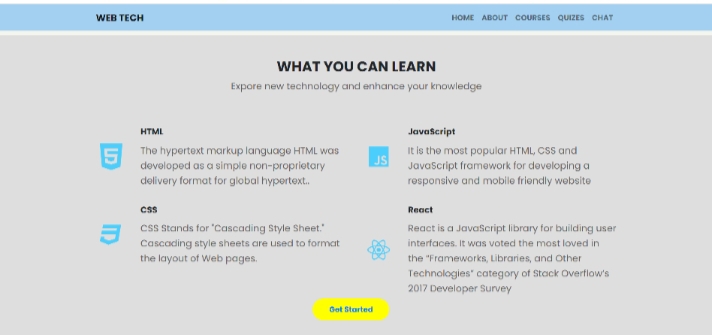
**4.2.3 SignUp page**



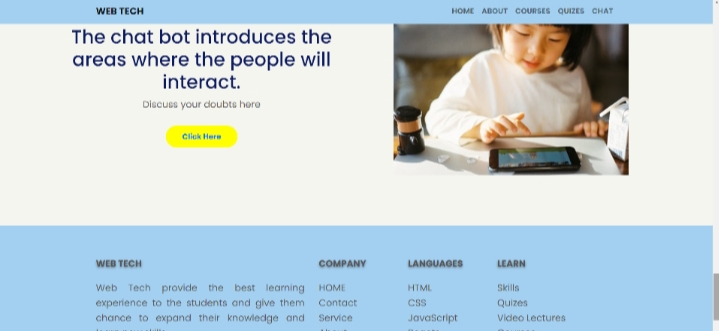
**4.2.4 home page**



**4.2.5 Overview page**



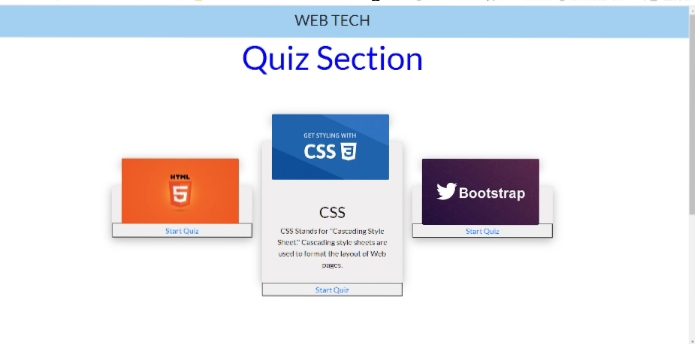
**4.2.6 Can Learn**



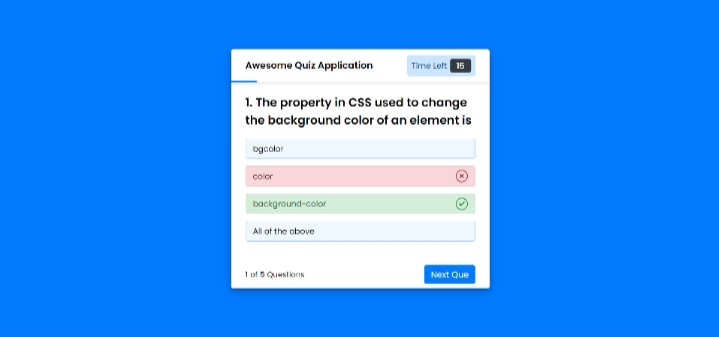
**4.2.7 Footer**



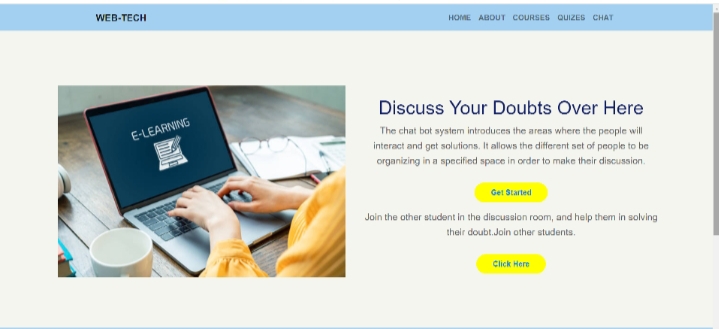
**4.2.8 Our Course**



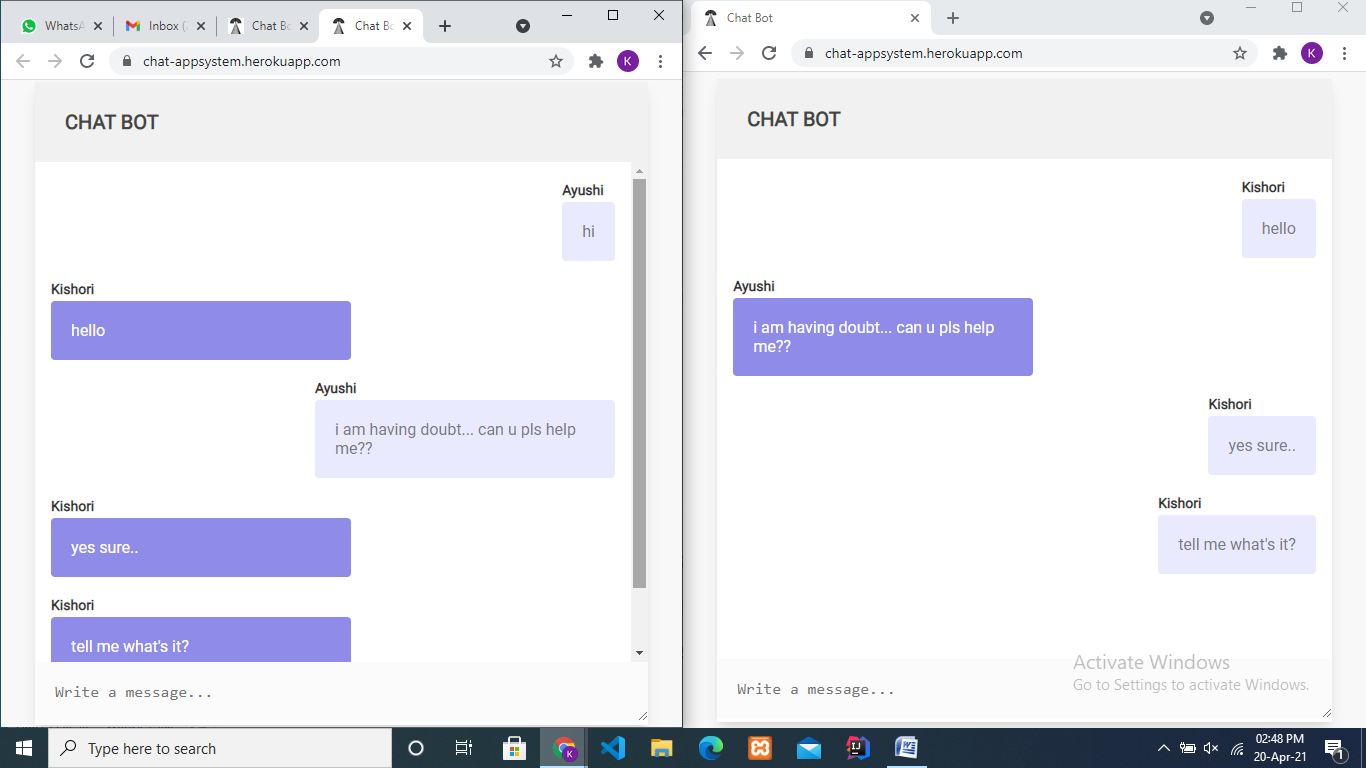
**4.2.9 Quiz Section**



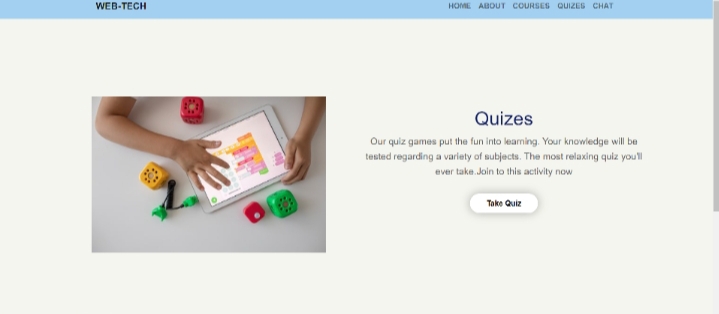
**4.2.10 Quiz Question**



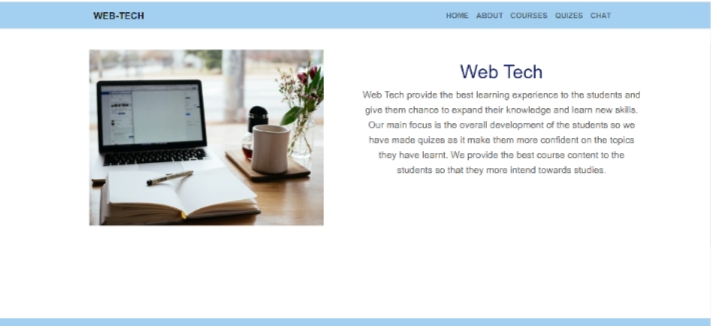
**4.2.11 Doubts page**



**4.2.13 Chat Bot**



**4.2.13 Quiz**



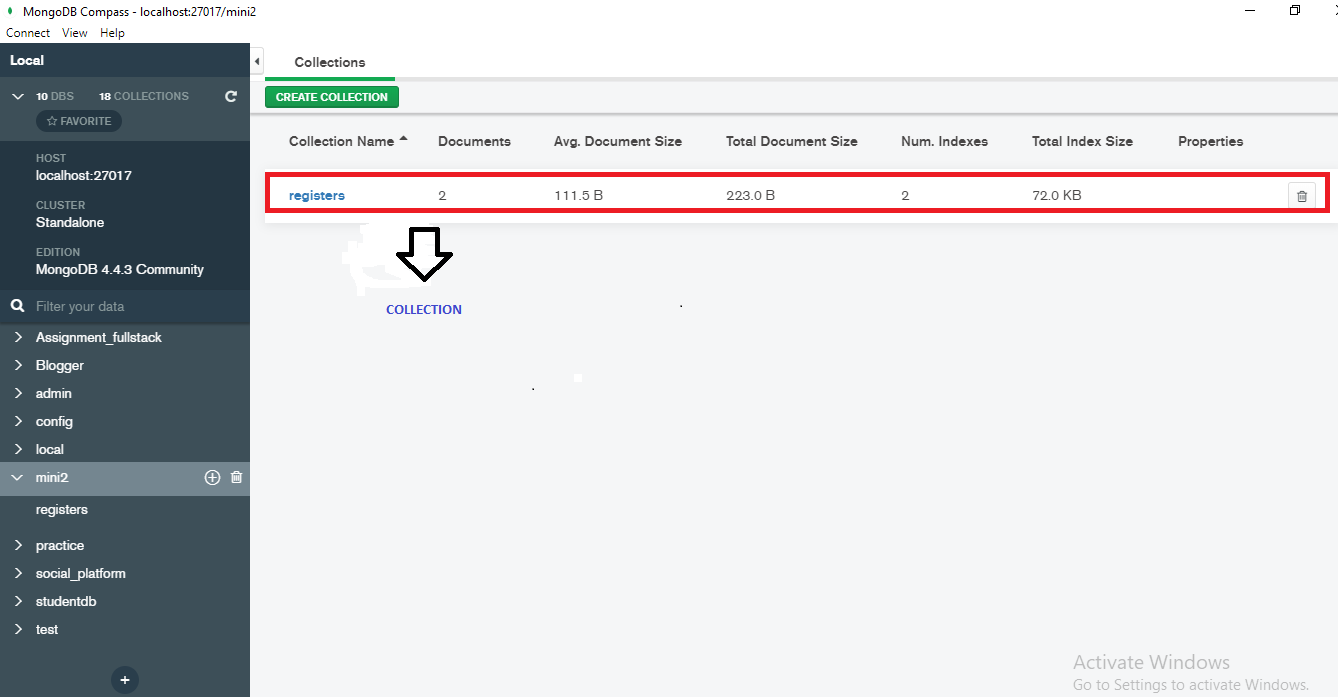
**4.2.14 About page**



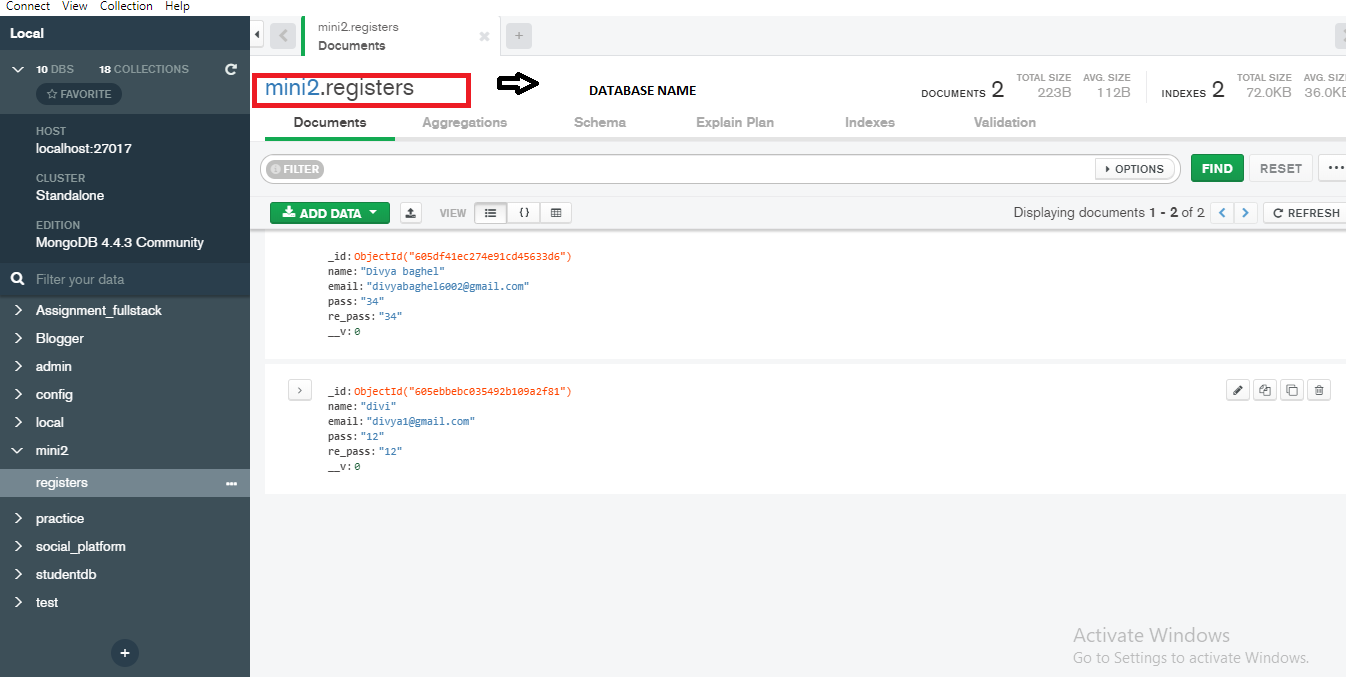
**4.2.15 Course Content**



**4.2.16 Content video**



**4.2.17 Database (Mongodb)**



**4.2.18 Database Connection in MongoDB.**

**Code :**

**1. Backend Code**

**App.js**

const { Console } = require("console");

const express = require("express");

const path =require("path");

const app = express();

//const hbs=require("hbs");

require("./db/conn");

const Register=require("./models/registers");

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

const static\_path = path.join(\_\_dirname,"../public");

const templates\_path =path.join(\_\_dirname,"../templates/views");

//const partials\_path =path.join(\_\_dirname,"../templates/partials");

app.use(express.static(static\_path));

app.set("view engine", "hbs");

app.set("views",templates\_path);

//hbs.registerPartials(partials\_path);

app.use(express.json());

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

app.get("/",(req,res)=>{

    res.render("login");

});

app.get("/index",(req,res)=>{

    res.render("index");

});

app.get("/register",(req,res)=>{

    res.render("register");

});

app.get("/about",(req,res)=>{

    res.render("about");

});

app.get("/chat",(req,res)=>{

    res.render("chat");

});

app.post("/register",async(req,res)=>{

    try{

        const password=req.body.pass;

        const cpassword=req.body.re\_pass;

        if(password===cpassword){

const registerEmployee =new Register({

    name:req.body.name,

    email:req.body.email,

    pass:req.body.pass,

    re\_pass:req.body.re\_pass,

})

const registered =await registerEmployee.save();

res.status(201).render("index");

        }else{

           res.render("register",{

          massage:"Password is not match",

           });

        }

    }

    catch(error){

        //res.status(400).send("Email already Exit");

        res.render("register",{

            massage:"Email already Exit",

             });

    }

})

app.post("/login",async(req,res)=>{

    try{

        const email=req.body.email;

        const password=req.body.pass;

    const useremail = await Register.findOne({email:email})

if(useremail.pass===password){

    res.status(201).render("index");

}

else{

    res.render("login",{

        massage:" Wrong Password ",

         });

}

    }

    catch(error){

       // res.status(400).send("invalid login Details");

       res.render("login",{

        massage:"invalid login Details",

         });

    }

})

app.listen(port ,() =>{

    console.log(`server is runing now ${port}`);

})

**Login.hbs**

<!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">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" />

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

      <link rel="stylesheet" href="/css/style1.css">

    <link rel="stylesheet" href="/css/style2.css">

    <title>Login</title>

</head>

<body>

 <div class="main">

      <nav class="navbar navbar-expand-md  navbar-light fixed-top" style="background-color: #e3f2fd;">

         <div class="container text-uppercase ps-2">

             <a class="navbar-brand" href="#">WEB TECH</a>

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

                 <span class="navbar-toggler-icon"></span>

             </button>

             <div class="collapse navbar-collapse" id="collapsibleNavbar">

                 <ul class="navbar-nav ml-auto">

                     <li class="nav-item">

                         <a class="nav-link" href="#">Home</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">About</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">Courses</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">Quizes</a>

                     </li>

                 </ul>

             </div>

         </div>

     </nav>

     <p class="mass">

   {{massage}}

</p>

  <section class="sign-in">

    <div class="container">

        <div class="signin-content">

            <div class="signin-image">

                <figure><img src="/css/img/signin-image.jpg" alt="sing up image"></figure>

                <a href="/register" class="signup-image-link">Create an account</a>

            </div>

            <div class="signin-form">

                <h2 class="form-title">Sign In</h2>

                <form method="POST" class="register-form" id="login-form" action="/login">

                    <div class="form-group">

                        <label for="your\_name"><i class="zmdi zmdi-account material-icons-name"></i></label>

                        <input type="text" name="email" id="your\_name" placeholder="Your Email" required/>

                    </div>

                    <div class="form-group">

                        <label for="your\_pass"><i class="zmdi zmdi-lock"></i></label>

                        <input type="password" name="pass" id="your\_pass" placeholder="Password" required/>

                    </div>

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

                        <input type="submit" name="signin" id="signin" class="form-submit" value="Log in"/>

                    </div>

                    <div>

                         <h6 class="forget">Forget password</h6>

                    </div>

                </form>

                <div class="social-login">

                    <span class="social-label">Or login with</span>

                    <ul class="socials">

                        <li><a href="#"><i class="display-flex-center zmdi-facebook fa fa-facebook"></i></a></li>

                        <li><a href="#"><i class="display-flex-center zmdi-twitter fa fa-twitter"></i></a></li>

                        <li><a href="#"><i class="display-flex-center zmdi-google fa fa-google"></i></a></li>

                    </ul>

                </div>

            </div>

        </div>

    </div>

</section>

 </div>

</body>

</html>

**Register.hbs:**

<!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>Sign Up</title>

    <link rel="stylesheet" href="fonts/material-icon/css/material-design-iconic-font.min.css">

 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

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

      <link rel="stylesheet" href="/css/style1.css">

    <link rel="stylesheet" href="/css/style2.css">

</head>

<body>

    <div class="main">

          <nav class="navbar navbar-expand-md  navbar-light fixed-top" style="background-color: #e3f2fd;">

         <div class="container text-uppercase ps-2">

             <a class="navbar-brand" href="#">WEB TECH</a>

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

                 <span class="navbar-toggler-icon"></span>

             </button>

             <div class="collapse navbar-collapse" id="collapsibleNavbar">

                 <ul class="navbar-nav ml-auto">

                     <li class="nav-item">

                         <a class="nav-link" href="#">Home</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">About</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">Courses</a>

                     </li>

                     <li class="nav-item">

                         <a class="nav-link" href="#">Quizes</a>

                     </li>

                 </ul>

             </div>

         </div>

     </nav>

     <p class="mass">

   {{massage}}

</p>

        <section class="signup">

            <div class="container">

                <div class="signup-content">

                    <div class="signup-form">

                        <h2 class="form-title">Sign Up</h2>

                        <form  action = "/register"  method="POST" class="register-form" id="register-form">

                            <div class="form-group">

                                <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>

                                <input type="text" name="name" id="name" placeholder="Your Name" required/>

                            </div>

                            <div class="form-group">

                                <label for="email"><i class="zmdi zmdi-email"></i></label>

                                <input type="email" name="email" id="email" placeholder="Your Email" required/>

                            </div>

                            <div class="form-group">

                                <label for="pass"><i class="zmdi zmdi-lock"></i></label>

                                <input type="password" name="pass" id="pass" placeholder="Password"/>

                            </div>

                            <div class="form-group">

                                <label for="re-pass"><i class="zmdi zmdi-lock-outline"></i></label>

                                <input type="password" name="re\_pass" id="re\_pass" placeholder="Repeat your password"/>

                            </div>

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

                                <input type="submit" name="signup" id="signup" class="form-submit" value="Register"/>

                            </div>

                        </form>

                    </div>

                    <div class="signup-image">

                        <figure><img src="/css/img/signup-image.jpg" alt="sing up image"></figure>

                        <p style="text-align: center;">or</p>

                        <a href="/" class="signup-image-link">Login</a>

                    </div>

                </div>

            </div>

        </section>

    </div>

</body>

</html>

**Mongoose**

**Registers.js**

const mongoose = require("mongoose");

const employeeSchema =new mongoose.Schema({

    name:{

        type:String,

        required :true

    },

    email:{

        type:String,

        required :true,

        unique:true

    },

    pass:{

         type:String,

          require:true

    },

    re\_pass:{

        type:String,

        require:true

    }

})

const Register=new mongoose.model("Register",employeeSchema);

module.exports=Register;

**CHAPTER 5**

**Software Testing**

Testing is one of the most important phases in the software development activity. In software development lifecycle, the main aim of testing process is the quality i.e the developed software is tested against maintaining the required functionality and performance.

During the testing process the software is worked with some particular test cases and the

output of the test cases are analyzed whether the software is working accordingly as expected or not.

**5.1 Introduction**

The success of the testing process in determining the errors mostly depends upon the test

case criteria. For testing any software we need to have a description of the expected behaviour of the system and the method of determining whether the observed behavior confirms to the expected behaviour or not.

**Levels of Testing**

Since the errors in the software can cause problems at any stage. So, we need to carry out the testing process at different level during the development. The basic levels of testing are unit, integration, system and acceptance testing.

**5.2 Testing Process**

A number of activities must be performed for testing software. Testing starts with test plan. Test plan identifies all testing that needs to be performed along with the schedule and guidelines.

**Test Plan:**

The test plan is general document for entire project, which defines the scope approach to be taken and the personal responsibilities for different activities of testing. The inputs for test plans are:

1. Project plan
2. Requirement document
3. System design

**Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

**System Testing**

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased .It includes both functional and Non-Functional testing

**REFERENCES**

1. <https://www.w3schools.com/>
2. <https://www.beta-labs.in/>
3. <https://stackoverflow.com/>
4. [www.nodejs.org](http://www.nodejs.org)
5. [www.monogodb.com](http://www.monogodb.com)
6. [www.youtube.com](http://www.youtube.com)
7. [www.bootstrap.com](http://www.bootstrap.com)