Certificate of Completion

**THIS ACKNOWLEDGES THAT**

**GODWIN ONYEDIKACHUKWU ODENIGBO**

**BONAVENTURE IGWE-AWEGBE**

**OSONDU WALLY NNAMANI**

**NNAMDI GODSON AZUBUIKE**

**HAS SUCCESSFULLY COMPLETED THE**

**DOCTORS WEB FORUM PROJECT**

**TABLE OF CONTENT**

ACKNOWLEDGEMENT - - - - - - **3**

**PROBLEM DEFINITION**

EPROJECT SYNOPSIS - - - - - - **4**

**CUSTOMER REQUIREMENT**

EPROJECT ANALYSIS - - - - - - **5**

**PROJECT SPECIFICATION AND PLAN**

EPROJECT DESIGN - - - - - **6**

**E-R DIAGRAMS AND ALGORITHMS**

DATA FLOW DIAGRAM - - - - - **7**

FLOW CHART - - - - - **8**

PROCESS DIAGRAM - - - - - **9**

DATABASE DESIGN AND STRUCTURE - - - **10**

**INTERFACE DESIGN**

SCREEN SHOT OF THE APPLICATION INTERFACE - **11**

GUI STANDARDS DOCUMENT  **- - -** - **25**

**DOCUMENT**

SOURCE CODE - - - - - 31

USER’S GUIDE - - - - - 17

DEVELOPER GUIDE - - - - - 128

MODULE DESCRIPTION - - - - - 29

TASK SHEET - - - - - 26

PROJECT REVIEW AND MONITORING REPORT - - 27

UNIT TESTING AND CHECK LIST` - - - - 28

**ACKNOWLEDGEMENT**

I on behalf of my team would like to thank God Almighty for his grace, blessings and inspirations throughout the challenging times on this e-project.

Also I would like to express my deep gratitude to Aptech as an institution which has stood as an academic giant in the ever evolving world of Information Technology and by that given I and so many students around the world a suitable, stable, and standard learning environment to unleash our potentials.

Again, I salute our dedicated center manager and our e-project for their patience, guidance, enthusiastic encouragement and constructive critiques to this project. I also wish to thank all our center faculties who have selflessly been great and exemplary teachers and guidance here in Aptech Enugu center.

Further, I wish to acknowledge my team and myself for the efforts, dedications, research, ideas and contributions we put together to make this project a success.

I would also like to express sincere thanks towards volunteer researchers who devoted their time and knowledge in gathering of research data and information which also contributed positively in the successful completion of this project.

My appreciation also goes to other students in our study center for sharing their insights and their constructive criticism which in all motivated us to putting our best effort to make this project stable, adoptable, usable, editable and accessible.

Finally, I express my gratitude towards our individual families and colleagues for their kind co-operation and encouragement which helped us too in the completion of this project within the limited time frame.

**Thanks.**

**EPROJECT SYNOPSIS**

Doctors web forum is a web based platform whose sole purpose is to provide its users whom are basically medical doctors an easy and efficient means of identification, communication, interaction, discussion and research.

This Doctors web forum is designed such that potential users whom are medical doctors can register on the platform using their contact details, personal details, professional profile, qualifications, experience, and as well their achievements in their medical career, thus creating an environment where upcoming doctors can learn from professional doctors and every user can engage and share research data or information.

Further, the doctor’s web forum is a platform where medical doctors can interact and share experiences as well ask questions from other doctors across various locations and specializations about emergencies, epidemic diseases, special cases and strange ailments. Also doctors who use this platform can discus medical device usage and the use and guide for newly introduced medications as well as sharing some safety tips.

Again, for every user on this forum, there would be provision to view some basic details of other doctors. To enable timely and continuous interactions and contributions, the system is designed such that the number of users online at each instance would be displayed on the forum for every user and as well there would be provision to hide some personal details from public view to enable the usage of the system flow well.

Generally, every user on this Doctors web forum has the basic permission to update his or her account details and can also only view the details of the other doctors or users on the platform, hide some personal details, post a query or a request, reply to a query or make a suggestion. While the admin has the permission to delete a user and perform other major operations that generally affect all the users on the platform.

**EPROJECT ANALYSIS**

Most times medical doctors come across an emergency situation, a new medical device or a strange ailment. At this point such a doctor may be a starter or a professional whom may specialize in one or many fields but such a situation maybe out of what current knowledge or experience such doctor has, at this point the doctor would need some information, research data or shared experience relating to that, such or similar ailment or professional guide to help provide immediate medical attention required to resuscitate the victim or remedy the situation in case of an outbreak or to understand the use of a medical device or medication newly introduced into the system.

In some situations, the doctor or doctors involved are forced to go on research, team work or data collection on various web platforms where they maybe misguided or misinformed, also they research across some regions where such outbreak has earlier been recorded. To gather correct research data, analyze, interpret, understand and make it a working medical document could take a very long time and mostly, in the medical field, every doctor often come across emergencies and due to pressure from such emergency situations some doctors may due to lack of suitable information or knowledge misuse a device, abuse a medication or misunderstand and as well misinterpret and outbreak leading to massive and continuous loss of funds, lives or deformities.

Further, this gave rise to the need for a doctors web forum where doctors from various parts, regions, fields, experiences and specializations would register and in a case of an emergency, newly introduced medical device, research or strange ailment the doctor, doctors or team involved can quickly login to the forum and make some descriptions as well ask questions and request the guide they need to attend to the respective situation while every other doctor on the platform would respond and contribution from various medical experiences would surely answer questions about the situation, medication, ailment or device involved.

**EPROJECT DESIGN**

This Doctors web forum is designed as a login platform with a basic front end to enable easy and user friendly landing interface where potential users can navigate through the web pages to get a view and understand what the forum is all about to help encourage users to register on the doctors web forum, the home page is designed with a bold quote at the top of the page “The Most Valuable Thing is Your Health” written in white and with a blue transparent background positioned directly on an image showing two doctors giving the user a beautiful and exciting welcome into the website. Immediately after the home page header is a basic category navigation showing major aspects of medicine and health care and each category on click is displayed beside the navigation side bar giving the user a cross view and easy exploration of the various medical categories described on the website, below that is a gallery view of some professional renowned doctors currently on the forum to build confidence of the user and maintain the reputation of the web forum, below the gallery view is a statistics block displaying the total number of doctors already registered in the web forum, the number of doctors active currently in the forum, the total number of questions asked and answered all to ensure every user who land on the website is fully convinced that the forum is a reliable giant for medical research or assistance next is a footer linking to other parts of the website.

Further are other web pages of the showing specified predefined where the about us discusses details of the web forum and the gallery shows a lot of medical images and users and within the web forum, next is the contact us providing contact information to enable enquiries, suggestions or contributions.

Then, the login platform is designed with a register button where on click a new user is provided a registration form and registered users are provided a login form. On login, users can post a question, reply or answer a question or suggestion and also perform changes on their profile or view other users profile while the admin can delete or remove a user, reply, comment or query.

**DATA FLOW DIAGRAM**

**CLICK OF REGISTER**

**REGISTER BUTTON**

**LOGIN**

**FORM**

**LOGIN**

**BUTTON**

**REGISTRATION**

**FORM**

**REGISTER BUTTON**

**LOGIN**

**FORM**

**DASHBOARD**

**VIEW QUERY**

**REPLIES**

**REPLY**

**QUERY**

**VIEW**

**QUERIES**

**POST**

**QUERY**

**END OF**

**FLOW**

**Image 1.0.**

**Data Flow Diagram of Doctors Web Forum.**

**FLOW CHAT**

**CONTACT US**

**ABOUT US**

**HOME**

**REGISTER**

**GALARY**

CONTACT

DETAILS

ABOUT FORUM

VITAL

DETAILS

FORUM

IMAGES

REGISTER

OR

LOGIN

**REGISTRATION**

**DETAILS**

If Register

If Login

**LOGIN**

**DETAILS**

**DASHBOARD**

POST OR

REPLY

QUERY

**ALL**

**QUERIES**

**VIEW**

**QUERIES**

**QUERY**

**REPLIES**

If post query

**ENTER**

**QUERY**

**VIEW QUERY**

**REPLIES**

If reply

query

**REPLY**

**QUERY**

**Image 1.1.**

**END OF FLOW**

**Flow Chat Diagram of Doctors Web Forum.**

**PROCESS DIAGRAM**

**HOME**

**LOGIN**

**REGISTER**

**DASHBOARD**

**VIEW QUERY**

**REPLIES**

**VIEW**

**QUERIES**

**POST**

**QUERY**

**POST QUERY**

**REPLY QUERIES**

**DISPLAY REPLIES**

**END OF PROCESS**

**Image 1.2.**

**Process Diagram of Doctors Web Forum.**

**DATABASE STRUCTURE**

|  |
| --- |
| **Users** |
| **Full name** |
| **Email** |
| **Password** |
| **Admin status** |
| **Achievements** |
| **Specialization** |
| **Indexes** |
| **User Id** |

|  |
| --- |
| **Queries** |
| **Query** |
| **Indexes** |
| **Query Id** |
| **User Id** |

<

|  |
| --- |
| **Response** |
| **Response** |
| **Indexes** |
| **Query Id** |
| **User Id** |

^

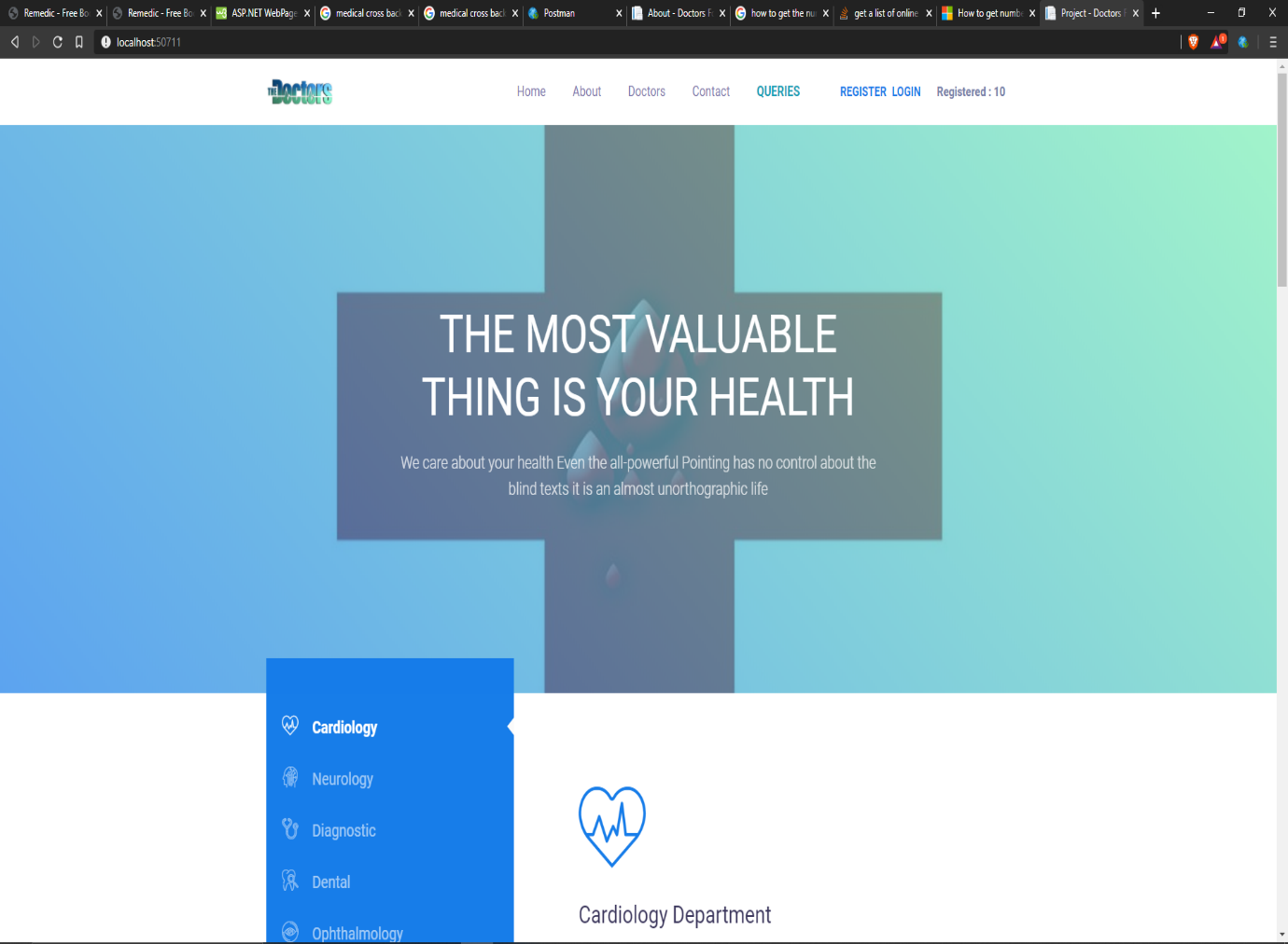
<

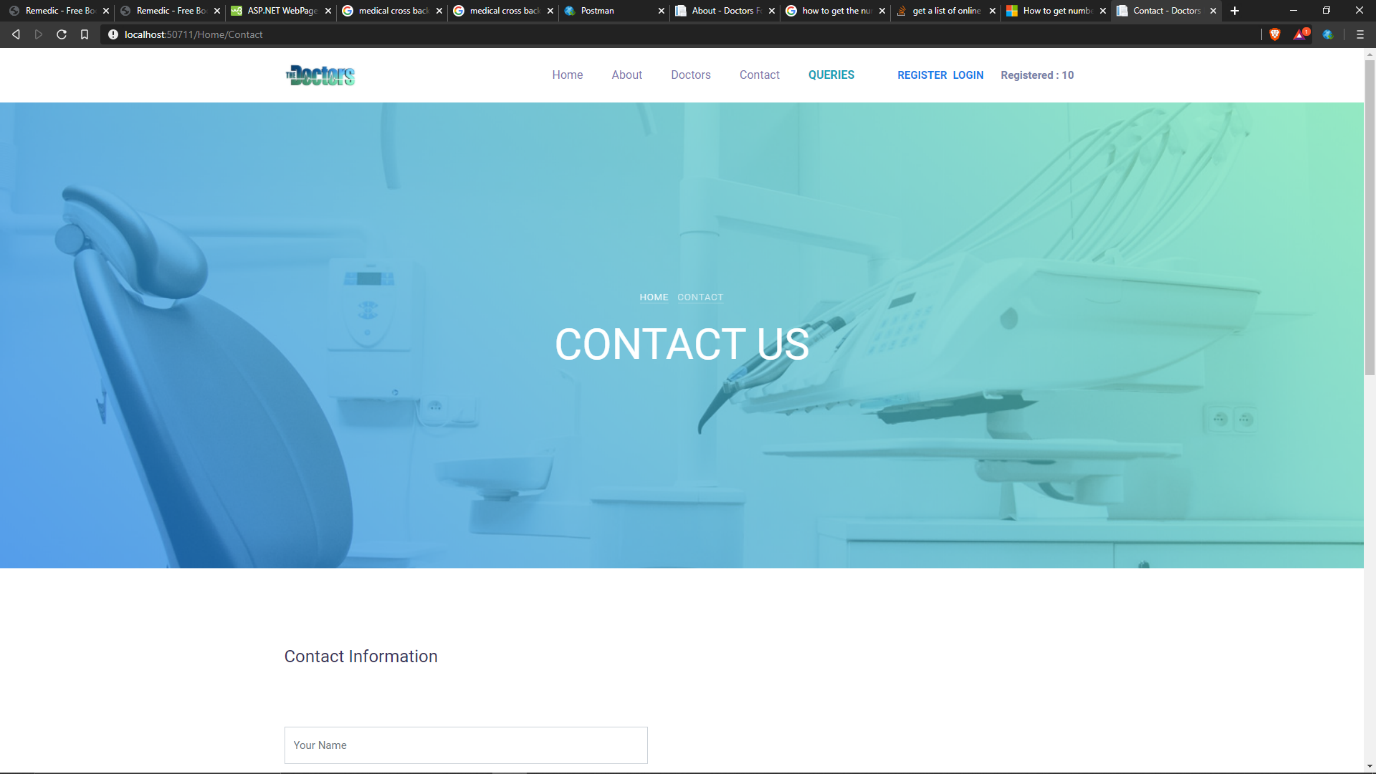
**Image 1.3.**

**Database Structure of Doctors Web Forum.**

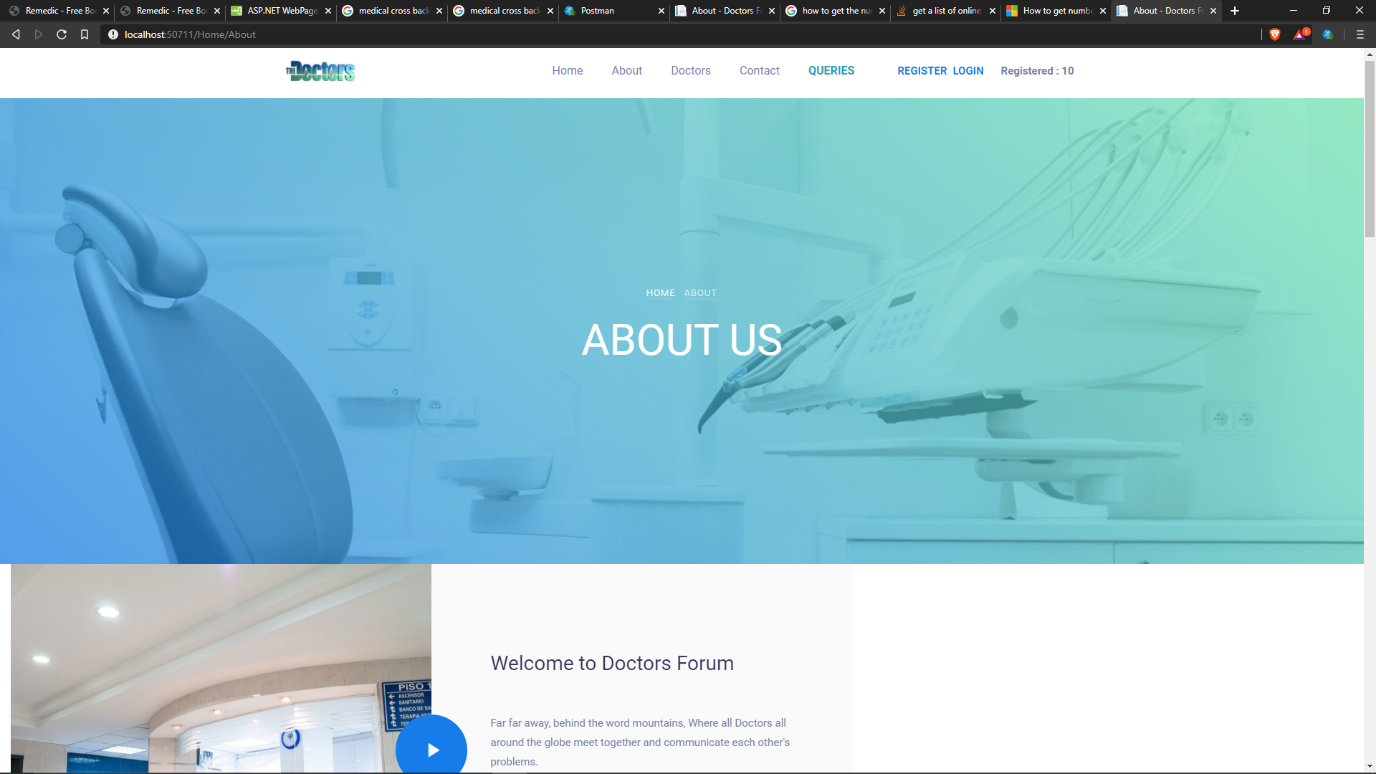
**WEB FORUM SCREENSHOTS**

**Home Page:**

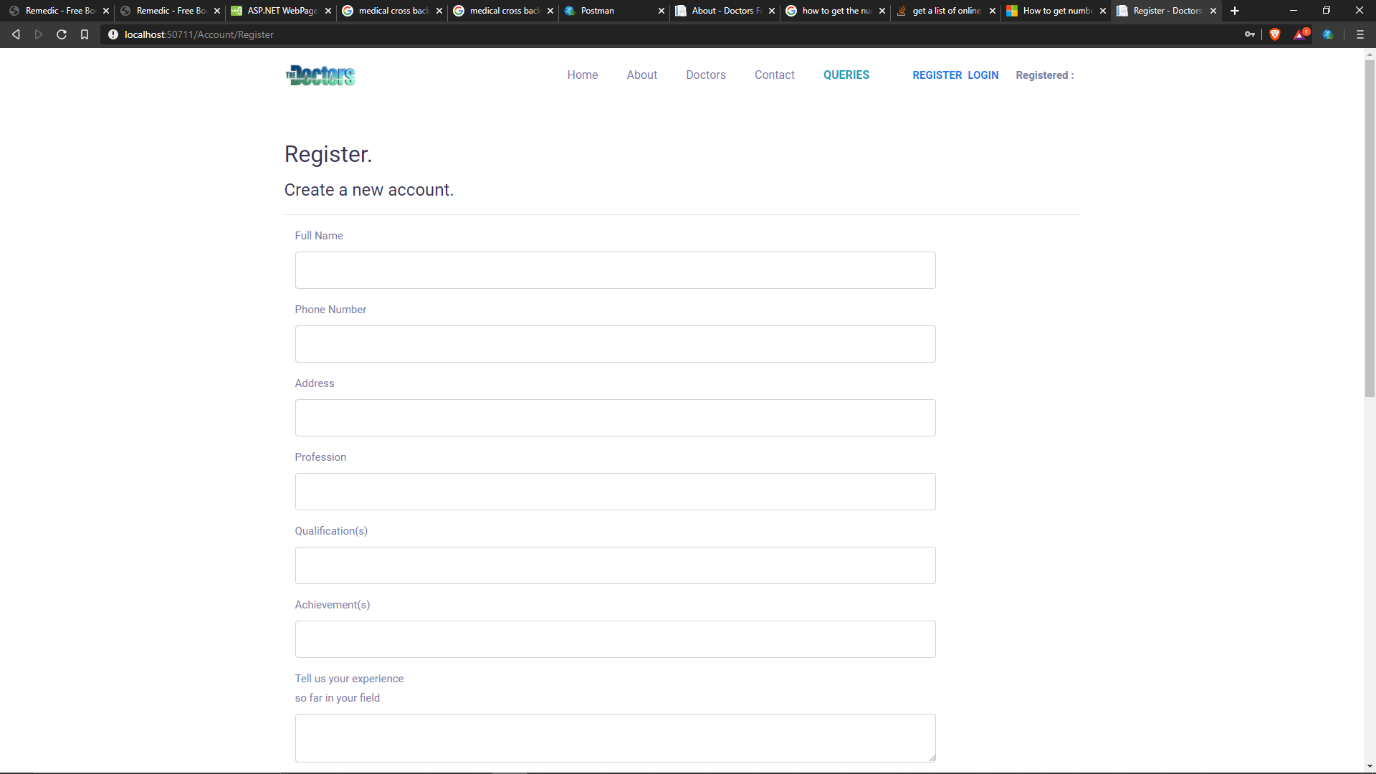
****

**Contact Us:**

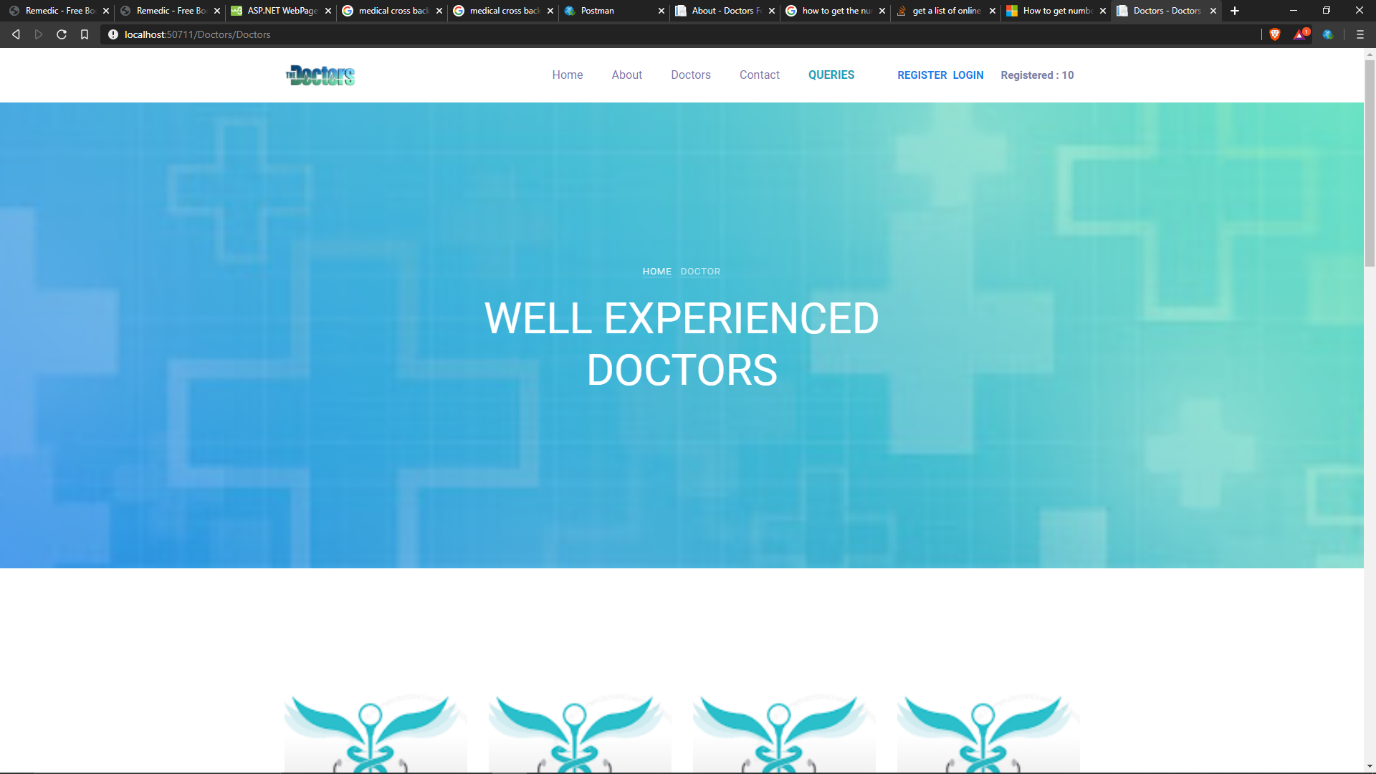
**About Us:**

****

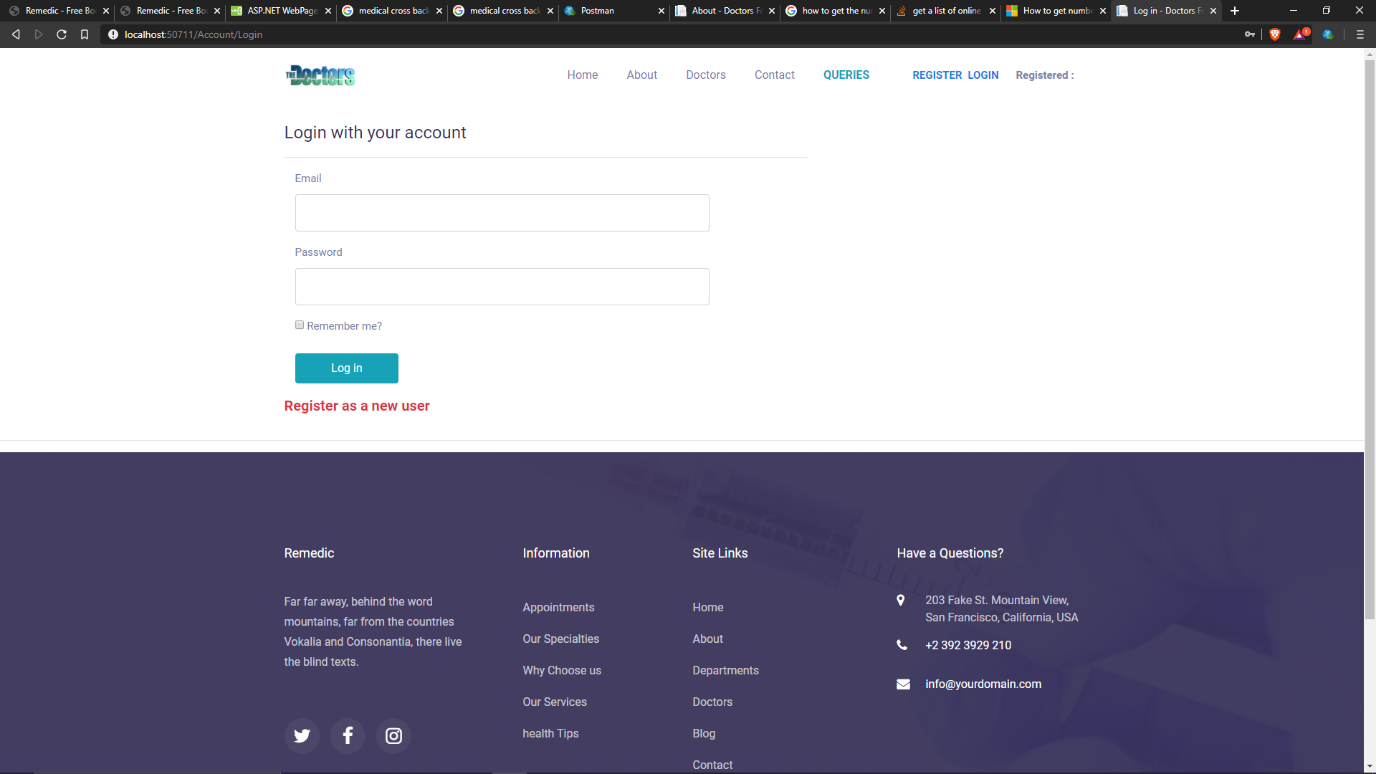
**Register:**

****

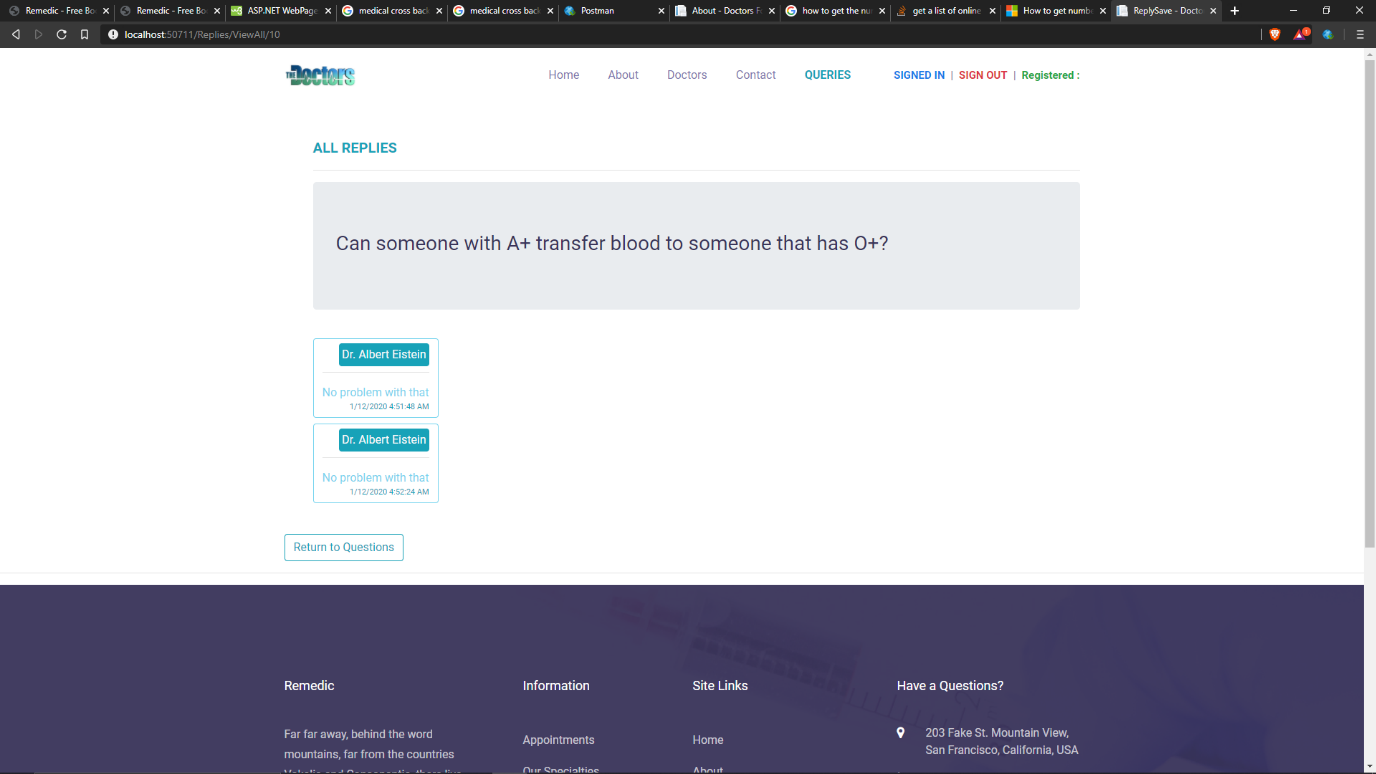
**Doctors:**

****

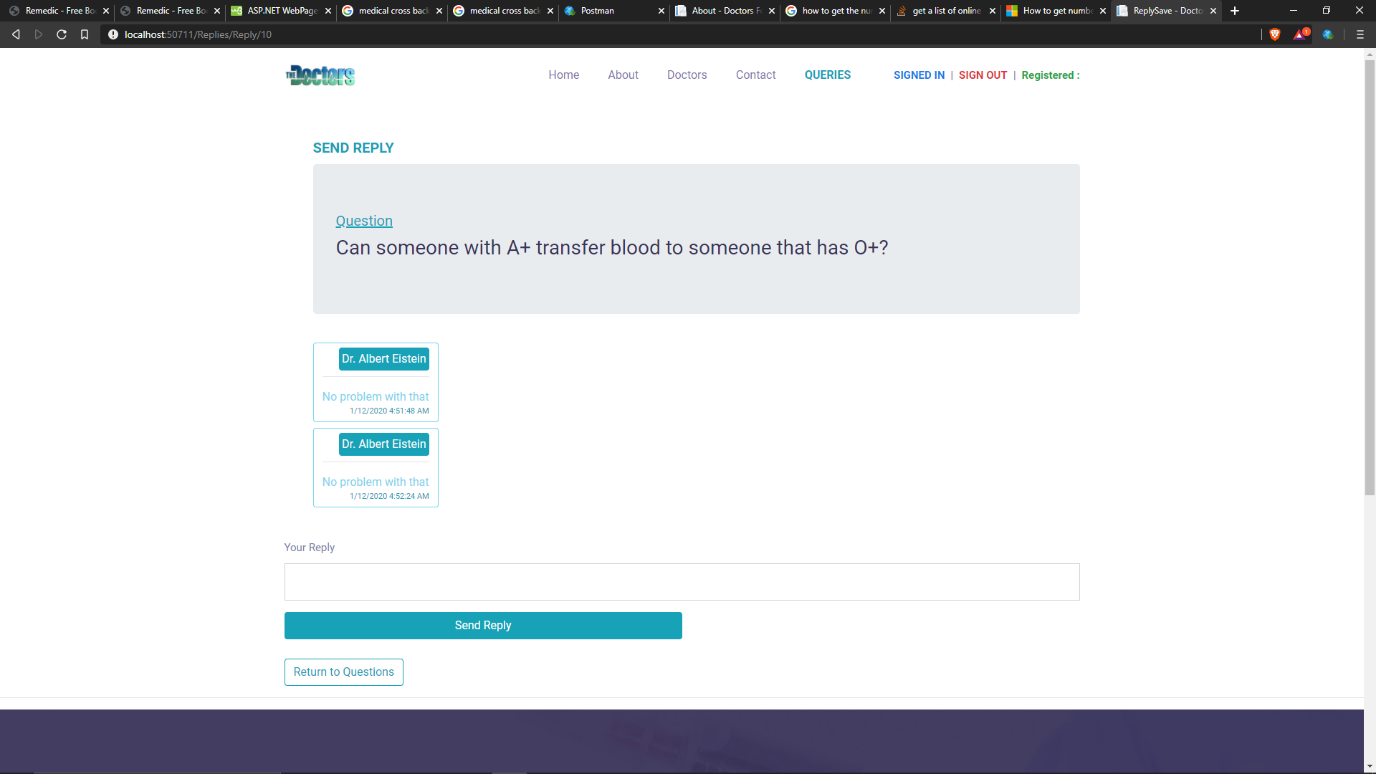
**Login:**

****

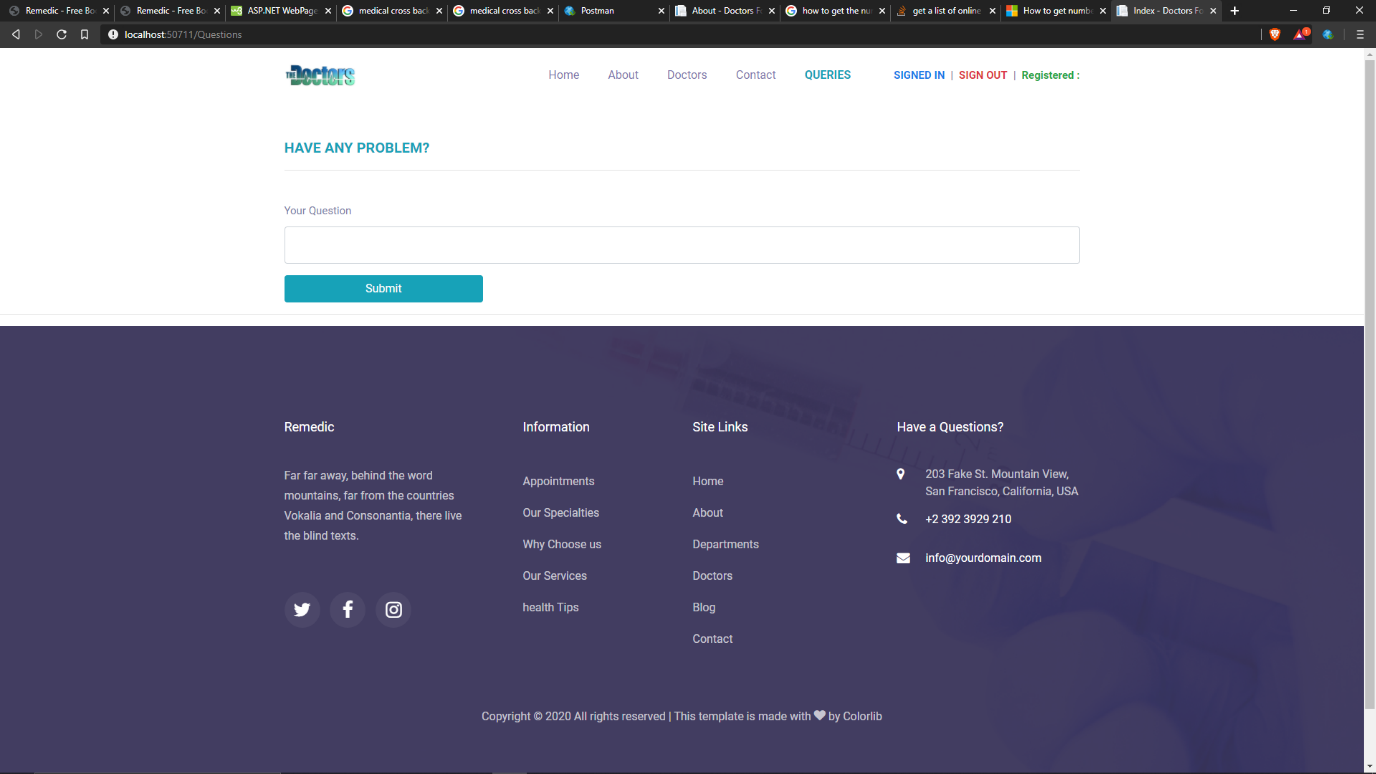
**All Reply:**

****

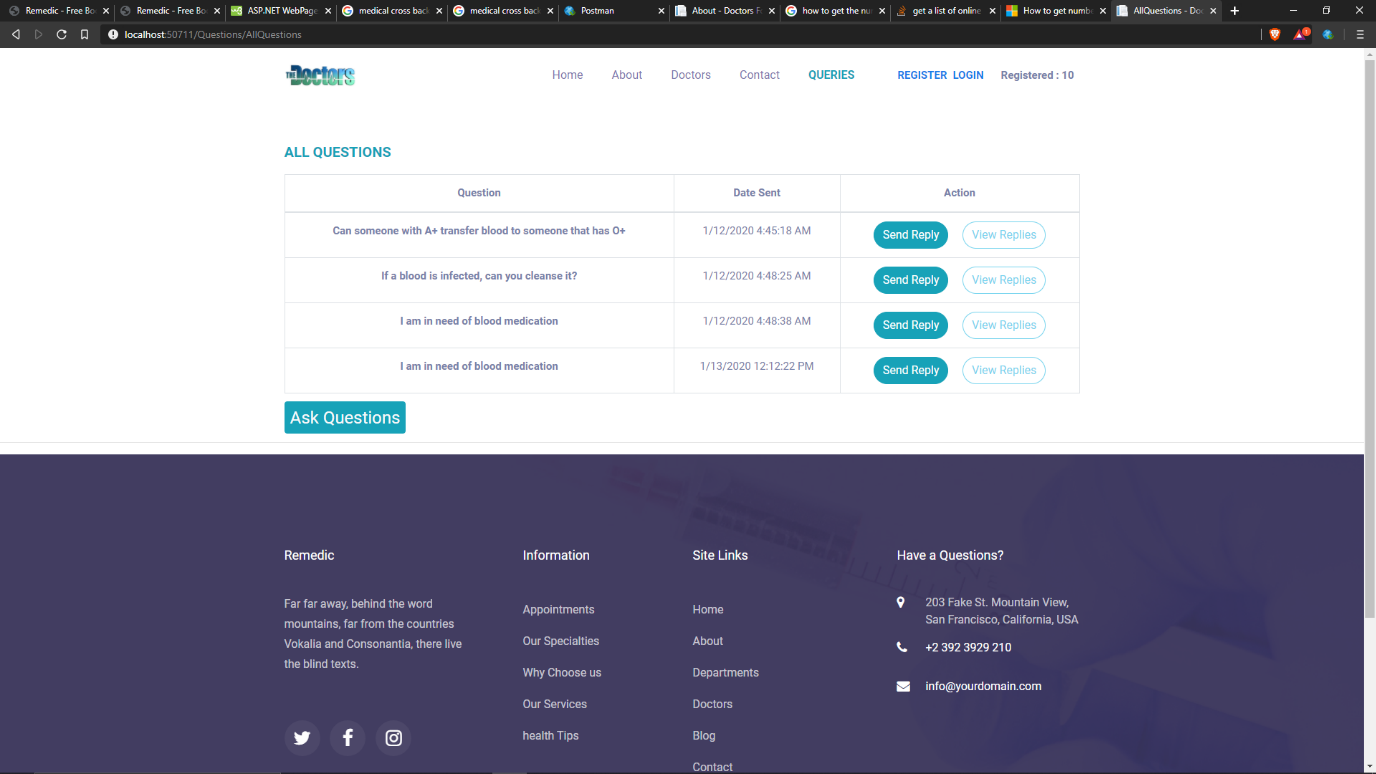
**Send Reply:**

****

**Ask Question:**

****

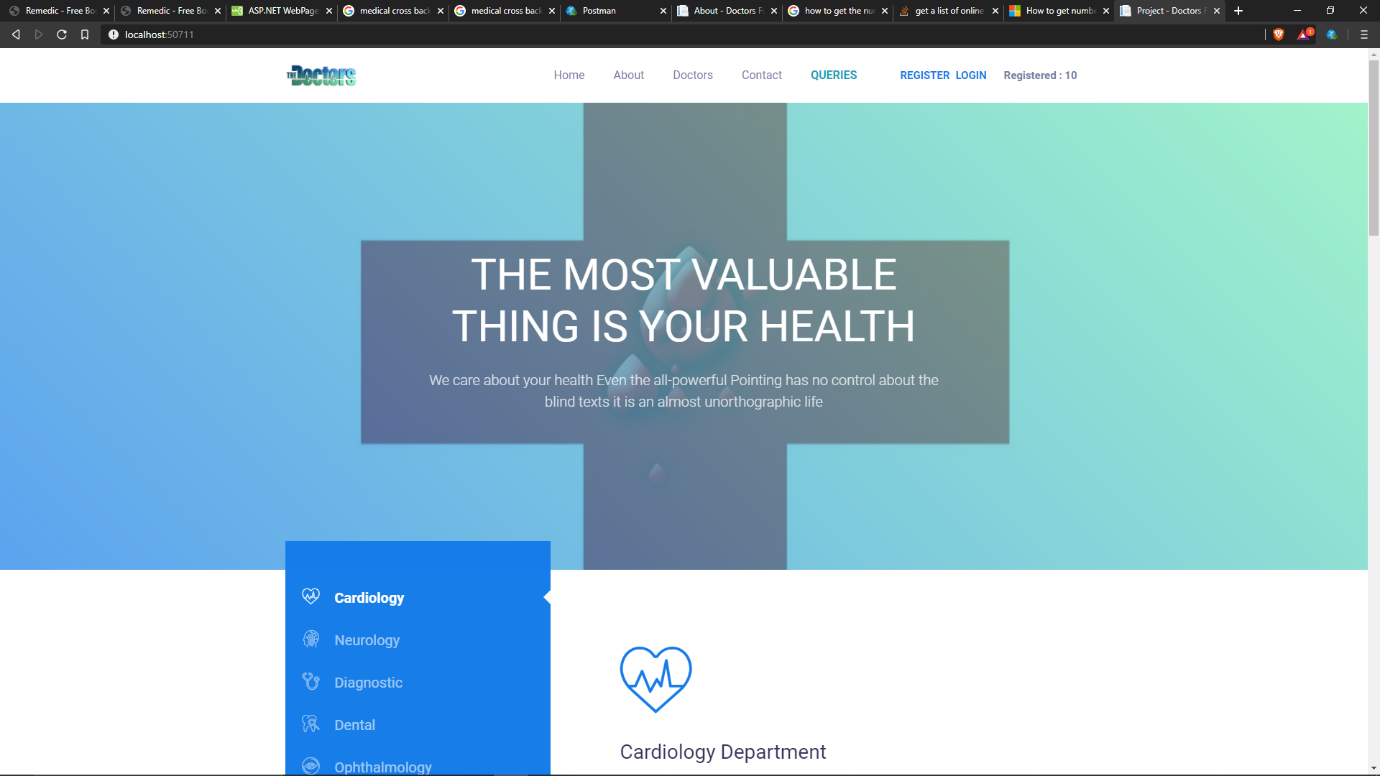
**All Questions:**

****

**WEB FORUM USER GUIDE**

* **User Registration:**

To register on this doctors web forum, at the website home page, navigate to register button at the top right corner of the webpage and click as shown in image 2.0, a registration form would appear, kindly provide all information required and click on the sign up button to complete registration as shown in image 2.1, on successful registration, a popup window would appear to show you have successfully registered on the platform.

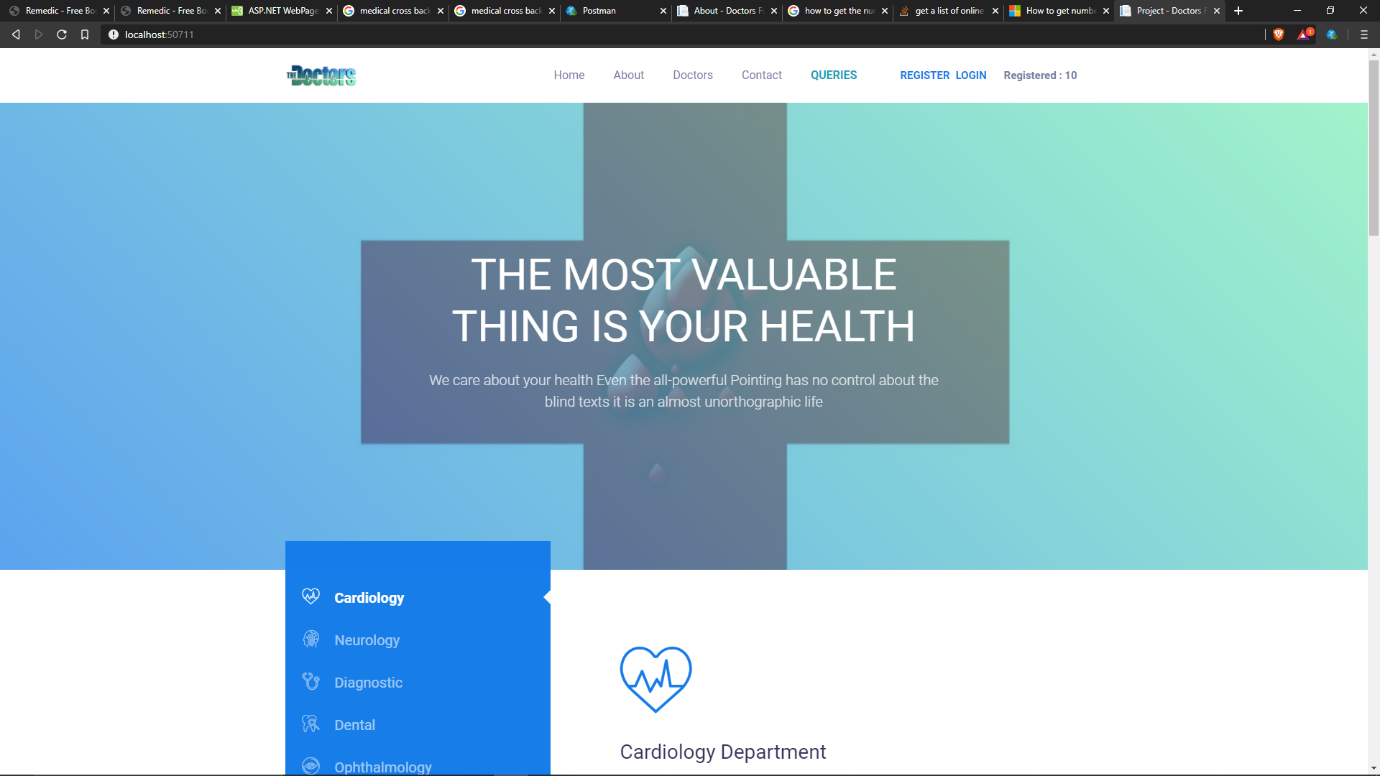
****

**Image 2.0.**

**Register Button.**

* **User Login:**

After Successful registration, a login form is visible at the registration page, kindly fill up the form and click on the login button to login to your doctor’s web forum account as shown in image 2.2.

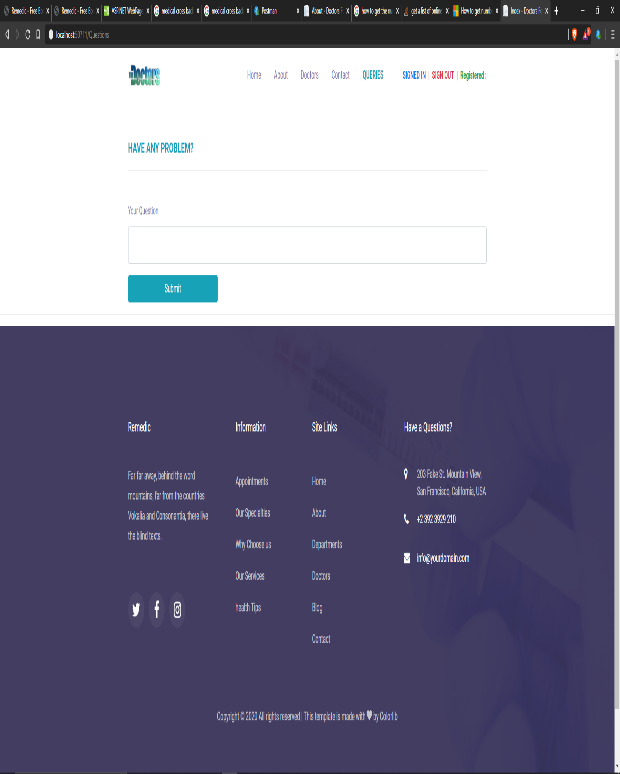
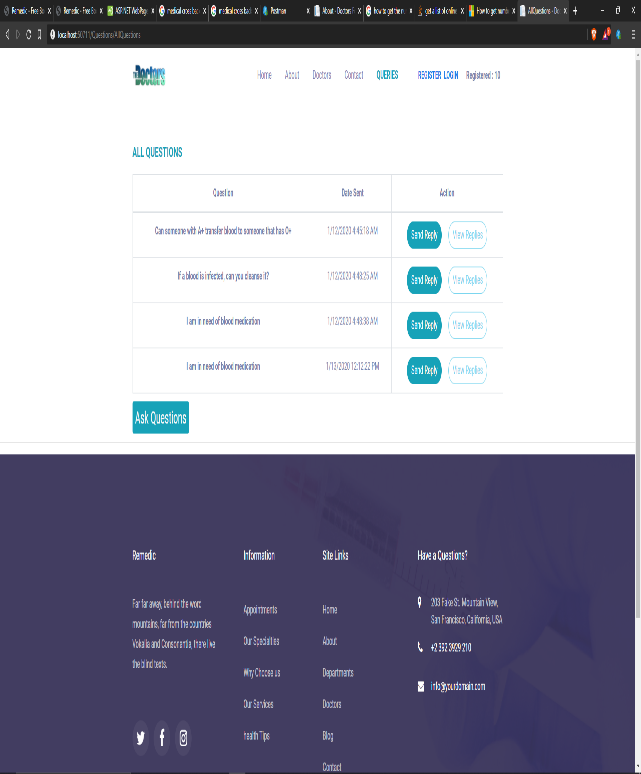
****

**Image 2.1.**

**Login Button.**

* **Post a Query:**

To post a query, navigate to Queries tab from your dashboard as in image 2.2, select ask question, type and post your question and you would be able to view the response as soon as they appear, refer to image 2.3 for details.

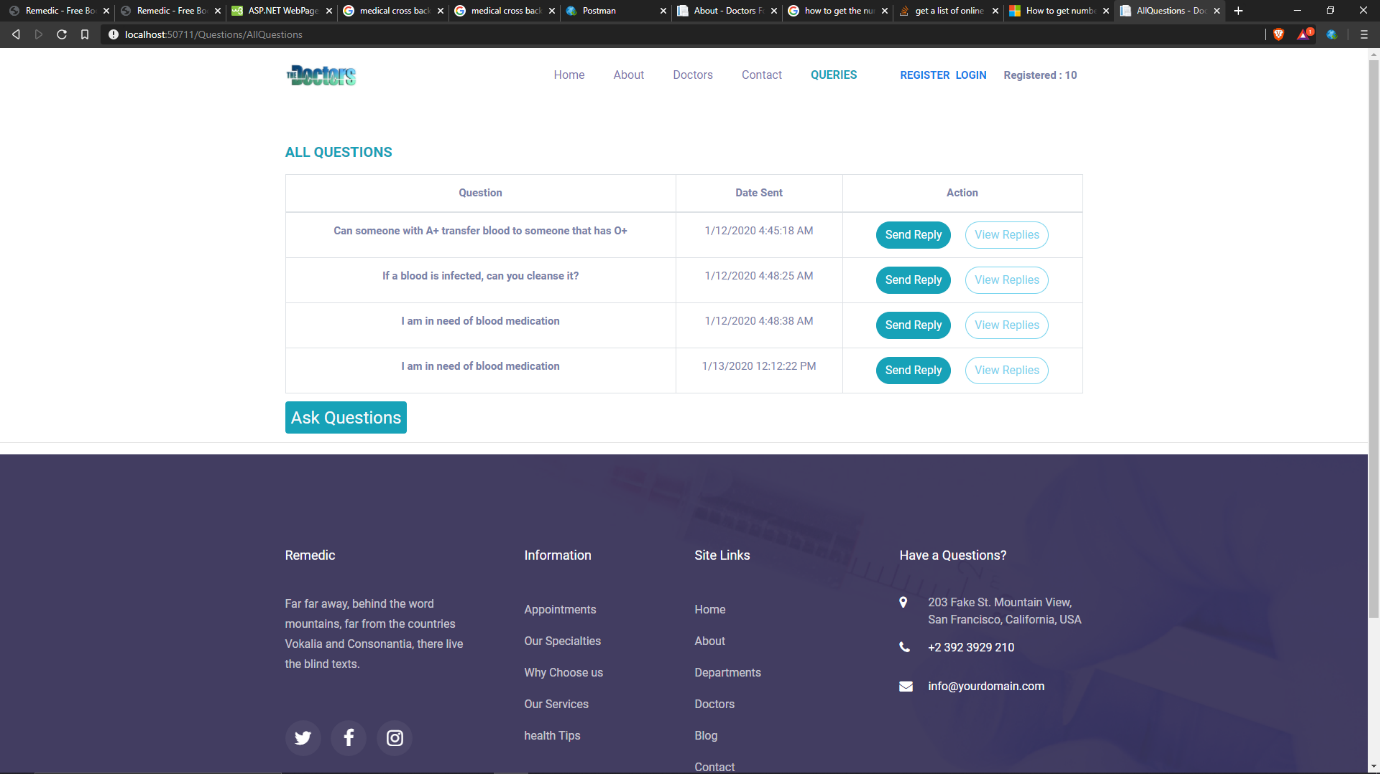


**Image 2.2. Image 2.3.**

**Post a Query. Ask Question.**

* **Answer a Query:**

To answer a query, navigate to the queries sections from your dashboard, a list of all the queries posted by other users would appear, select the question you choose to answer and click on answer question, provide your answer and post as shown in image 2.4.



**Image 2.4.**

**Answer a Query.**

**GUI STANDARD DOCUMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **RULES** | **APPLICATION** | **VIOLATION** |
| 1 | Visibility | Contents of this project are designed with readable and suitable images, formats, font styles and font sizes. | Not Violated |
| 2 | Accessibility | Proper use of navigation links and responsive media queries were observed to ensure easy system accessibility. | Not Violated |
| 3 | Consistency | Basic background design and color were maintained across all pages of this project to ensure consistency in design. | Not Violated |
| 4 | Realistic | Images, terms and symbols used in this project are physically associated with the real world use of this project. | Not Violated |
| 5 | Feedback | This project is interactively designed with popup windows and tooltips to ensure easy flow in the use of this program. | Not Violated |
| 6 | Design Standard | Universal symbols, Icons and images were properly used for their respective denotation to maintain design standard. | Not Violated |
| 7 | Hierarchy | This program is designed to flow in a user friendly hierarchy to ensure new users are able to explore the forum. | Not Violated |
| 8 | Simplicity | Throughout this project, content delivery and excellent user interface was at high priority to ensure a simply interface. | Not Violated |
| 9 | Flexibility | This forum is designed by progressive enhancement to ensure future updates will not make the system strange. | Not Violated |
| 10 | Documentation | Every part of this project is properly documented to enable new users find help and also provide possible future update. | Not Violated |

**EPROJECT TASKSHEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **TASK** | **STATUS** | **DURATION**  **(HOURS)** | **PERCENTAGE**  **(COMPLETED)** |
| 1 | A common logo and a cross background on all the pages of the doctors forum. | DONE | 1 | 100% |
| 2 | Online registration, login, contacting and posting/replying of questions and queries. | DONE | 8 | 100% |
| 3 | Display of the total number of users and the total number of currently active users. | DONE | 1 | 100% |
| 4 | Implementing a pop up window is to appear if a user registers or if the user enters a wrong password. | DONE | 1 | 100% |
| 5 | Increasing the total number of users on the forum once a new user registers. | DONE | 1 | 100% |
| 6 | Enabling users display or hide their profile information if they desire. | DONE | 1 | 100% |
| 7 | Collection of users contact details, personal, professional, qualification and achievement details during registration. | DONE | 8 | 100% |
| 8 | Implementation of an advanced search option that includes location, specialization, and experience. | DONE | 2 | 100% |
| 9 | Creation of a separate login for the admin and a different login for all other users. | DONE | 2 | 100% |
| 10 | Restricting users to only be able to update or access their own account details, view other user’s details, post or reply a query. | DONE | 2 | 100% |
| 11 | Enabling the Administrator remove or deleting any other user on the forum. | DONE | 3 | 100% |
|  | TOTAL DURATION AND STATUS | DONE | 30 | 100% |

**PROJECT REVIEW AND MONITORING REPORT**

Based on the information provided on the task sheet used for this project design and implementation, this project review and monitoring confirms the following functionality.

**Standard landing home page, descriptive web pages, consistency in design and standard design rules:**

This review understands and confirms that the project described in the project design has a home page and other web pages and navigation links describing what the forum stands form, about the forum and providing medium of contacting the management of the forum, further, the project is designed to display categories of medical science where users of the forum are professionals as well the forum has a gallery where images of devices, medications and users who are currently on the forum is displayed, this project compliance to GUI standard design has also been confirmed.

**Online registration, login, contacting and posting/replying of questions and queries:**

From the information collected during review of the doctors web forum, it has been confirmed that the forum if deployed would be able to successfully hold online registrations for its potential users, username and password protected login has also been confirmed, registered users can navigate through their dashboard to make post of queries or make suggestions, contributions or answer questions, users can view previous questions, the user who asked the question and the replies to the question with their respective responders, a user can access his profile, make changes to his personal details, hide his profile or make it visible to other users while admin users can further remove or delete any user and having confirmed all this features this project has met its basic aim and is hence considered stable for deployment to end users.

**UNIT TESTING AND CHECK LIST**

**Home Page:** The home page is working excellently well with adequate content delivery and excellent user experience.

**About Us:** The About us page is working excellently well with adequate content delivery and excellent user experience.

**Contact Us:** The Contact us page is working excellently well with adequate content delivery and excellent user experience.

**Gallery:** The Gallery page is working excellently well with adequate content delivery and excellent user experience.

**Register:** The user registration feature is working excellently well with adequate engagement and excellent user experience.

**Login:** The user login feature is working excellently well with adequate security and excellent user experience.

**Post Query:** Registered users logged in can effortlessly post queries, questions or suggestions on the forum without hitches.

**Reply Query:** Registered users logged in can effortlessly reply to queries, questions or suggestions on the forum without hitches.

**Edit Profile:** Registered users logged in on this web forum can edit their personal profile details, show or hide their profile from public view.

**Admin Function:** An admin on this doctors web forum can remove or delete a user and such function is only limited to an admin.

**Module Description**

Modules in DoctorsControllers:

public DoctorsController() : This is the public constructor for the Doctors Controller. the Application database context was Initialized here.

public ActionResult Doctors() : it is a public function that returns a View with a list of all the doctors registered in this application.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor to avoid database openings.

Modules in QuestionsControllers:

public QuestionsController() : This is the public constructor for the Questions Controller. the Application database context was Initialized here.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor.

public ActionResult Index() : this is a public function that returns the view for the questions Page.

public ActionResult Add(Questions questions) : This is a public function that adds a new question submitted by a user in the application. it takes an Object parameter of Question Model Class then saves it in the Database.

public ActionResult AllQuestions() : it is a public function that returns a View with a list of all the questions asked by users in this application.

Modules in RepliesControllers:

public RepliesController() : This is the public constructor for the Replies Controller. the Application database context was Initialized here.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor.

public ActionResult Index() : it is a public function that returns a View with a list of all the replies for a particular question asked by a user in this application.

public ActionResult Reply(int id) : It is a public function with a parameter Id of type Integer used to get the question then returns a view with the question gotten by the parameter passed and with a list of all the replies for the question.

public ActionResult ReplySave(Replies replies, int id): It is a public function with a parameter of type Object of the Replies Model Class and id of the question. it is used to save replies for a particular Question and returns a view with the question and a list of replies for the question.

public ActionResult ViewAll(int id): It is a public function with a parameter Id of type Integer used to get all the replies for a particular Question. It returns a view with the list of replies for the question.

SOURCE CODE

**Index Cshtml**

@model IEnumerable<SimpleChat.Models.ApplicationUser>

@{

ViewBag.Title = "Project";

}

<div class="hero-wrap" style="background-image: url('../../assets/images/red-cross-4411127\_\_340.jpg '); background-position: center; background-size: contain;">

<div class="overlay"></div>

<div class="container">

<div class="row no-gutters slider-text align-items-center justify-content-center" data-scrollax-parent="true">

<div class="col-md-8 ftco-animate text-center">

<h1 class="mb-4">The most valuable thing is your Health</h1>

<p>We care about your health Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life</p>

</div>

</div>

</div>

</div>

<section class="ftco-services">

<div class="container">

<div class="row no-gutters">

<div class="col-md-4 ftco-animate py-5 nav-link-wrap">

<div class="nav flex-column nav-pills" id="v-pills-tab" role="tablist" aria-orientation="vertical">

<a class="nav-link px-4 active" id="v-pills-master-tab" data-toggle="pill" href="#v-pills-master" role="tab" aria-controls="v-pills-master" aria-selected="true"><span class="mr-3 flaticon-cardiogram"></span> Cardiology</a>

<a class="nav-link px-4" id="v-pills-buffet-tab" data-toggle="pill" href="#v-pills-buffet" role="tab" aria-controls="v-pills-buffet" aria-selected="false"><span class="mr-3 flaticon-neurology"></span> Neurology</a>

<a class="nav-link px-4" id="v-pills-fitness-tab" data-toggle="pill" href="#v-pills-fitness" role="tab" aria-controls="v-pills-fitness" aria-selected="false"><span class="mr-3 flaticon-stethoscope"></span> Diagnostic</a>

<a class="nav-link px-4" id="v-pills-reception-tab" data-toggle="pill" href="#v-pills-reception" role="tab" aria-controls="v-pills-reception" aria-selected="false"><span class="mr-3 flaticon-tooth"></span> Dental</a>

<a class="nav-link px-4" id="v-pills-sea-tab" data-toggle="pill" href="#v-pills-sea" role="tab" aria-controls="v-pills-sea" aria-selected="false"><span class="mr-3 flaticon-vision"></span> Ophthalmology</a>

<a class="nav-link px-4" id="v-pills-spa-tab" data-toggle="pill" href="#v-pills-spa" role="tab" aria-controls="v-pills-spa" aria-selected="false"><span class="mr-3 flaticon-ambulance"></span> Emergency</a>

</div>

</div>

<div class="col-md-8 ftco-animate p-4 p-md-5 d-flex align-items-center">

<div class="tab-content pl-md-5" id="v-pills-tabContent">

<div class="tab-pane fade show active py-5" id="v-pills-master" role="tabpanel" aria-labelledby="v-pills-master-tab">

<span class="icon mb-3 d-block flaticon-cardiogram"></span>

<h2 class="mb-4">Cardiology Department</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

<div class="tab-pane fade py-5" id="v-pills-buffet" role="tabpanel" aria-labelledby="v-pills-buffet-tab">

<span class="icon mb-3 d-block flaticon-neurology"></span>

<h2 class="mb-4">Neurogoly Department</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

<div class="tab-pane fade py-5" id="v-pills-fitness" role="tabpanel" aria-labelledby="v-pills-fitness-tab">

<span class="icon mb-3 d-block flaticon-stethoscope"></span>

<h2 class="mb-4">Diagnostic Department</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

<div class="tab-pane fade py-5" id="v-pills-reception" role="tabpanel" aria-labelledby="v-pills-reception-tab">

<span class="icon mb-3 d-block flaticon-tooth"></span>

<h2 class="mb-4">Dental Departments</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

<div class="tab-pane fade py-5" id="v-pills-sea" role="tabpanel" aria-labelledby="v-pills-sea-tab">

<span class="icon mb-3 d-block flaticon-vision"></span>

<h2 class="mb-4">Ophthalmology Departments</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

<div class="tab-pane fade py-5" id="v-pills-spa" role="tabpanel" aria-labelledby="v-pills-spa-tab">

<span class="icon mb-3 d-block flaticon-ambulance"></span>

<h2 class="mb-4">Emergency Departments</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nesciunt voluptate, quibusdam sunt iste dolores consequatur</p>

<p>Inventore fugit error iure nisi reiciendis fugiat illo pariatur quam sequi quod iusto facilis officiis nobis sit quis molestias asperiores rem, blanditiis! Commodi exercitationem vitae deserunt qui nihil ea, tempore et quam natus quaerat doloremque.</p>

</div>

</div>

</div>

</div>

</div>

</section>

<section class="ftco-section-2 img" style="background-image: url(../../assets/images/bg\_3.jpg);">

<div class="container">

<div class="row d-md-flex justify-content-end">

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

<div class="row">

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

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>Laboratory Services</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>General Treatment</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>Emergency Service</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>24/7 Help &amp; Support</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

</div>

</div>

</div>

</div>

</div>

</section>

<section class="ftco-section">

<div class="container">

<div class="row justify-content-center mb-5 pb-3">

<div class="col-md-7 heading-section ftco-animate text-center">

<h2 class="mb-4">Our Experienced Doctors</h2>

</div>

</div>

<div class="row">

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style="background-image: url(../../assets/images/doctor-1.jpg);">

<div class="box">

<h2></h2>

<p>Neurologist</p>

</div>

</div>

<div class="back">

<!-- back content -->

<blockquote>

<p>&ldquo;Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doctor-1.jpg);"></div>

</div>

<div class="name align-self-center">Aldin Powell <span class="position">Neurologist</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

</div>

<div class="row">

<div class="col-md-9 ftco-animate">

<h4>We are well experienced doctors</h4>

<p>A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.</p>

</div>

</div>

</div>

</section>

<section class="ftco-section ftco-counter img" id="section-counter" style="background-image: url(../../assets/images/bg\_4.jpg);">

<div class="container">

<div class="row justify-content-center mb-5 pb-3">

<div class="col-md-7 text-center heading-section heading-section-white ftco-animate">

<h2 class="mb-4">Some fun facts</h2>

<span class="subheading">Far far away, behind the word mountains, far from the countries</span>

</div>

</div>

<div class="row justify-content-center">

<div class="col-md-10">

<div class="row">

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="60">0</strong>

<span>Hospital</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="200">0</strong>

<span>Doctors</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="100">0</strong>

<span>Clinics</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="200">0</strong>

<span>Reviews</span>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- loader -->

<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px" height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke="#eeeeee" /><circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-miterlimit="10" stroke="#F96D00" /></svg></div>

**ABOUT.CSHTML**

@model SimpleChat.Models.Questions

@{

ViewBag.Title = "About";

}

<div class="hero-wrap" style="background-image: url('../../assets/images/bg\_6.jpg'); background-attachment:fixed;">

<div class="overlay"></div>

<div class="container">

<div class="row no-gutters slider-text align-items-center justify-content-center" data-scrollax-parent="true">

<div class="col-md-8 ftco-animate text-center">

<p class="breadcrumbs"><span class="mr-2"><a href="index.html">Home</a></span> <span>About</span></p>

<h1 class="mb-3 bread">About Us</h1>

</div>

</div>

</div>

</div>

<section class="ftco-section-2">

<div class="container-fluid d-flex">

<div class="section-2-blocks-wrapper row no-gutters">

<div class="img col-sm-12 col-lg-6" style="background-image: url('../../assets/images/about.jpg');">

<a href="https://vimeo.com/45830194" class="button popup-vimeo"><span class="ion-ios-play"></span></a>

</div>

<div class="text col-lg-6 ftco-animate">

<div class="text-inner align-self-start">

<h3>Welcome to Doctors Forum</h3>

<p>Far far away, behind the word mountains, Where all Doctors all around the globe meet together and communicate each other's problems.</p>

<p>A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.</p>

</div>

</div>

</div>

</div>

</section>

<section class="ftco-section">

<div class="container">

<div class="row justify-content-center mb-5 pb-3">

<div class="col-md-7 heading-section ftco-animate text-center">

<h2 class="mb-4">Our Experienced Doctors</h2>

</div>

</div>

<div class="row">

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style="background-image: url(../../assets/images/doctor-1.jpg);">

<div class="box">

<h2>Aldin Powell</h2>

<p>Neurologist</p>

</div>

</div>

<div class="back">

<!-- back content -->

<blockquote>

<p>&ldquo;Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doctor-1.jpg);"></div>

</div>

<div class="name align-self-center">Aldin Powell <span class="position">Neurologist</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style="background-image: url(../../assets/images/doctor-2.jpg);">

<div class="box">

<h2>Aldin Powell</h2>

<p>Pediatrician</p>

</div>

</div>

<div class="back">

<!-- back content -->

<blockquote>

<p>&ldquo;Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doctor-2.jpg);"></div>

</div>

<div class="name align-self-center">Aldin Powell <span class="position">Pediatrician</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style="background-image: url(../../assets/images/doctor-3.jpg);">

<div class="box">

<h2>Aldin Powell</h2>

<p>Ophthalmologist</p>

</div>

</div>

<div class="back">

<!-- back content -->

<blockquote>

<p>&ldquo;Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doctor-3.jpg);"></div>

</div>

<div class="name align-self-center">Aldin Powell <span class="position">Ophthalmologist</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style="background-image: url(../../assets/images/doctor-4.jpg);">

<div class="box">

<h2>Aldin Powell</h2>

<p>Pulmonologist</p>

</div>

</div>

<div class="back">

<!-- back content -->

<blockquote>

<p>&ldquo;Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doctor-4.jpg);"></div>

</div>

<div class="name align-self-center">Aldin Powell <span class="position">Pulmonologist</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

</div>

<div class="row">

<div class="col-md-9 ftco-animate">

<h4>We are well experienced doctors</h4>

<p>A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.</p>

</div>

</div>

</div>

</section>

<section class="ftco-section-2 img" style="background-image: url(../../assets/images/bg\_3.jpg);">

<div class="container">

<div class="row d-md-flex justify-content-end">

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

<div class="row">

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

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>Laboratory Services</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>General Treatment</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>Emergency Service</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

<a href="#" class="services-wrap ftco-animate">

<div class="icon d-flex justify-content-center align-items-center">

<span class="ion-ios-arrow-back"></span>

<span class="ion-ios-arrow-forward"></span>

</div>

<h2>24/7 Help &amp; Support</h2>

<p>Even the all-powerful Pointing has no control about the blind.</p>

</a>

</div>

</div>

</div>

</div>

</div>

</section>

<section class="ftco-section ftco-counter img mt-2" id="section-counter" style="background-image: url(../../assets/images/bg\_4.jpg);">

<div class="container">

<div class="row justify-content-center mb-5 pb-3">

<div class="col-md-7 text-center heading-section heading-section-white ftco-animate">

<h2 class="mb-4">Some fun facts</h2>

<span class="subheading">Far far away, behind the word mountains, far from the countries</span>

</div>

</div>

<div class="row justify-content-center">

<div class="col-md-10">

<div class="row">

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="60">0</strong>

<span>Hospital</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="200">0</strong>

<span>Doctors</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="100">0</strong>

<span>Clinics</span>

</div>

</div>

</div>

<div class="col-md-3 d-flex justify-content-center counter-wrap ftco-animate">

<div class="block-18 text-center">

<div class="text">

<strong class="number" data-number="200">0</strong>

<span>Reviews</span>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- loader -->

<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px" height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke="#eeeeee" /><circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-miterlimit="10" stroke="#F96D00" /></svg></div>

<!-- Modal -->

<div class="modal fade mt-5" id="modalAppointment" tabindex="-1" role="dialog" aria-labelledby="modalAppointmentLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalAppointmentLabel">Appointment</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

@using (Html.BeginForm("Add", "Questions"))

{

<div class="form-group">

@Html.LabelFor(m => m.QuestionText)

@Html.TextBoxFor(m => m.QuestionText, new { @class = "form-control" })

</div>

@Html.HiddenFor(m => m.UserId)

<button type="submit" class="btn btn-primary">Submit</button>

}

</div>

</div>

</div>

</div>

**Contact.cshtml**

@model SimpleChat.Models.Questions

@{

ViewBag.Title = "Contact";

}

<div class="hero-wrap" style="background-image: url('../../assets/images/bg\_6.jpg'); background-attachment:fixed;">

<div class="overlay"></div>

<div class="container">

<div class="row no-gutters slider-text align-items-center justify-content-center" data-scrollax-parent="true">

<div class="col-md-8 ftco-animate text-center">

<p class="breadcrumbs"><span class="mr-2"><a href="index.html">Home</a></span> <span>Contact</span></p>

<h1 class="mb-3 bread">Contact Us</h1>

</div>

</div>

</div>

</div>

<section class="ftco-section contact-section ftco-degree-bg">

<div class="container">

<div class="row d-flex mb-5 contact-info">

<div class="col-md-12 mb-4">

<h2 class="h4">Contact Information</h2>

</div>

<div class="w-100"></div>

</div>

<div class="row block-9">

<div class="col-md-6 pr-md-5">

<form action="#">

<div class="form-group">

<input type="text" class="form-control" placeholder="Your Name">

</div>

<div class="form-group">

<input type="text" class="form-control" placeholder="Your Email">

</div>

<div class="form-group">

<input type="text" class="form-control" placeholder="Subject">

</div>

<div class="form-group">

<textarea name="" id="" cols="30" rows="7" class="form-control" placeholder="Message"></textarea>

</div>

<div class="form-group">

<input type="submit" value="Send Message" class="btn btn-primary py-3 px-5">

</div>

</form>

</div>

<div class="col-md-6" id="map"></div>

</div>

</div>

</section>

<!-- loader -->

<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px" height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke="#eeeeee" /><circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-miterlimit="10" stroke="#F96D00" /></svg></div>

<!-- Modal -->

<div class="modal fade mt-5" id="modalAppointment" tabindex="-1" role="dialog" aria-labelledby="modalAppointmentLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalAppointmentLabel">Appointment</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

@using (Html.BeginForm("Add", "Questions"))

{

<div class="form-group">

@Html.LabelFor(m => m.QuestionText)

@Html.TextBoxFor(m => m.QuestionText, new { @class = "form-control" })

</div>

@Html.HiddenFor(m => m.UserId)

<button type="submit" class="btn btn-primary">Submit</button>

}

</div>

</div>

</div>

</div>

**REPLY.CSHTML**

@model SimpleChat.chatModel.ViewReplies

@{

ViewBag.Title = "ReplySave";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container mt-5">

<ul>

<h5 class="font-weight-bold text-info">SEND REPLY</h5>

@foreach (var ques in Model.Questions)

{

<div class="jumbotron">

<span class="text-info h5"><u>Question</u></span>

<h3>@ques.QuestionText? </h3>

</div>

}

</ul>

<ul>

@if (!Model.Reply.Any())

{

<li class="btn btn-info btn-block btn-lg">No replies yet! <br /> Be the first</li>

}

@foreach (var reply in Model.Reply)

{

<li class="btn btn-default rounded mt-2 btn-outline-primary">

<span class="float-right bg-info p-1 rounded text-light">@reply.User.FullName</span> <br /> <hr />

@reply.RepliesText <br />

<span style="font-size: 11px;" class="text-info float-right">@reply.ReplyDate</span>

</li>

<br />

}

</ul>

<br />

<div>

@foreach (var qId in Model.Questions)

{

using (Html.BeginForm("ReplySave", "Replies", new { id = qId.Id }))

{

<div class="form-group">

@Html.LabelFor(m => m.Replies.RepliesText)

@Html.TextBoxFor(m => m.Replies.RepliesText, new { @class = "form-control" })

</div>

<button type="submit" class="rounded w-50 text btn btn-info">Send Reply</button>

}

}

<br />

@Html.ActionLink("Return to Questions", "AllQuestions", "Questions", null, new { @class = "btn rounded btn-outline-info w-20" })

</div>

</div>

**ReplySave.cshtml**

@model SimpleChat.chatModel.ViewReplies

@{

ViewBag.Title = "ReplySave";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Replies</h2>

<ul>

@foreach (var ques in Model.Questions)

{

<div class="jumbotron"><h3>@ques.QuestionText?</h3></div>

}

</ul>

<ul>

@foreach (var reply in Model.Reply)

{

<li class="btn btn-default" style="margin-top: 10px; text-align: right;">

<span class="text-danger">@reply.User.FullName</span> <br />

@reply.RepliesText <br />

<span style="color: lightslategray">@reply.ReplyDate</span>

</li>

<br />

}

</ul>

<div style="margin-left: 700px;">

<h3>@Html.ActionLink("Return to Questions", "AllQuestions", "Questions")</h3>

</div>

**VIEWALL.CSHTML**

@model SimpleChat.chatModel.ViewReplies

@{

ViewBag.Title = "ReplySave";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container mt-5">

<ul>

<h5 class="font-weight-bold text-info">ALL REPLIES</h5> <hr />

@foreach (var ques in Model.Questions)

{

<div class="jumbotron">

<h3>@ques.QuestionText?</h3>

</div>

}

</ul>

<ul>

@if (!Model.Reply.Any())

{

<li class="btn btn-info btn-block btn-lg">No replies yet! <br /> Be the first</li>

}

@foreach (var reply in Model.Reply)

{

<li class="btn btn-default rounded mt-2 btn-outline-primary">

<span class="float-right bg-info p-1 rounded text-light">@reply.User.FullName</span> <br /> <hr />

@reply.RepliesText <br />

<span style="font-size: 11px;" class="text-info float-right">@reply.ReplyDate</span>

</li>

<br />

}

</ul>

<br />

@Html.ActionLink("Return to Questions", "AllQuestions", "Questions", null, new { @class = "btn rounded btn-outline-info w-20" })

</div>

**ALLQUESTIONS.CSHTML**

@model IEnumerable<SimpleChat.Models.Questions>

@{

ViewBag.Title = "AllQuestions";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<br /><br />

@if (!Model.Any())

{

<div class="jumbotron"><h3 class="text-danger">No Questions Yet!!</h3></div>

}

else

{

<div class="container">

<table class="table text-center table-bordered table-hover mt-3">

<h5 class="font-weight-bold text-info">ALL QUESTIONS</h5>

<thead>

<tr>

<th>Question</th>

<th>Date Sent</th>

<th>Action</th>

</tr>

</thead>

<tbody>

@foreach (var question in Model)

{

<tr>

<td><b>@question.QuestionText</b></td>

<td>@question.DateSent</td>

<td>

@Html.ActionLink("Send Reply", "Reply", "Replies", new { id = question.Id }, new { @class = "btn btn-info mr-3" })

@Html.ActionLink("View Replies", "ViewAll", "Replies", new { id = question.Id }, new { @class = "btn btn-outline-primary" })

</td>

</tr>

}

</tbody>

</table>

<h4>@Html.ActionLink("Ask Questions", "Index", "Questions", null, new { @class = "bg-info text-light p-2 rounded" })</h4>

</div>

}

**INDEX.CSHTML (QUESTIONS CONTROLLER)**

@model SimpleChat.Models.Questions

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container mt-5">

<h5 class="font-weight-bold text-info">HAVE ANY PROBLEM?</h5>

<hr /> <br />

@using (Html.BeginForm("Add", "Questions"))

{

<div class="form-group">

@Html.LabelFor(m => m.QuestionText)

@Html.TextBoxFor(m => m.QuestionText, new { @class = "form-control rounded" })

</div>

@Html.HiddenFor(m => m.UserId)

<button type="submit" class="btn btn-info w-25 rounded">Submit</button>

}

</div>

**DOCTORS.CSHTML**

@model IEnumerable<SimpleChat.Models.ApplicationUser>

@{

ViewBag.Title = "Doctors";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="hero-wrap" style="background-image: url('../../assets/images/bgbg (2).jpg'); background-attachment:fixed;">

<div class="overlay"></div>

<div class="container">

<div class="row no-gutters slider-text align-items-center justify-content-center" data-scrollax-parent="true">

<div class="col-md-8 ftco-animate text-center">

<p class="breadcrumbs"><span class="mr-2"><a href="index.html">Home</a></span> <span>Doctor</span></p>

<h1 class="mb-3 bread">Well Experienced Doctors</h1>

</div>

</div>

</div>

</div>

<section class="ftco-section">

<div class="container">

<div class="row">

@foreach (var user in Model)

{

<div class="col-md-6 col-lg-3 ftco-animate">

<div class="block-2">

<div class="flipper">

<div class="front" style=" background-image: url(../../assets/images/doc.png);

">

<div class="box">

<h2>@user.FullName</h2>

<p>@user.Profession</p>

</div>

</div>

<div class="back">

<h6 class="p-2 text-info font-weight-bold">EXPERIENCE</h6>

<!-- back content -->

<blockquote>

<p>&ldquo;@user.Experience&rdquo;</p>

</blockquote>

<div class="author d-flex">

<div class="image mr-3 align-self-center">

<div class="img" style="background-image: url(../../assets/images/doc.png);"></div>

</div>

<div class="name align-self-center">@user.FullName <span class="position">@user.Profession</span></div>

</div>

</div>

</div>

</div> <!-- .flip-container -->

</div>

}

</div>

@\*<div class="row mt-5">

<div class="col text-center">

<div class="block-27">

<ul>

<li><a href="#">&lt;</a></li>

<li class="active"><span>1</span></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

<li><a href="#">4</a></li>

<li><a href="#">5</a></li>

<li><a href="#">&gt;</a></li>

</ul>

</div>

</div>

</div>\*@

</div>

</section>

<section class="ftco-section-parallax">

<div class="parallax-img d-flex align-items-center">

<div class="container">

<div class="row d-flex justify-content-center">

<div class="col-md-7 text-center heading-section heading-section-white ftco-animate">

<h2>Subcribe to our Newsletter</h2>

<p>Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in</p>

<div class="row d-flex justify-content-center mt-5">

<div class="col-md-8">

<form action="#" class="subscribe-form">

<div class="form-group d-flex">

<input type="text" class="form-control" placeholder="Enter email address">

<input type="submit" value="Subscribe" class="submit px-3">

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- loader -->

<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px" height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke="#eeeeee" /><circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-miterlimit="10" stroke="#F96D00" /></svg></div>

**\_LAYOUT.CSHTML**

<!DOCTYPE html>

<html lang="en">

<head>

<title>@ViewBag.Title - Doctors Forum</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,700" rel="stylesheet">

<link rel="stylesheet" href="~/Content/css/open-iconic-bootstrap.min.css">

<link rel="stylesheet" href="~/Content/css/animate.css">

<link rel="stylesheet" href="~/Content/css/owl.carousel.min.css">

<link rel="stylesheet" href="~/Content/css/owl.theme.default.min.css">

<link rel="stylesheet" href="~/Content/css/magnific-popup.css">

<link rel="stylesheet" href="~/Content/css/aos.css">

<link rel="stylesheet" href="~/Content/css/ionicons.min.css">

<link rel="stylesheet" href="~/Content/css/bootstrap-datepicker.css">

<link rel="stylesheet" href="~/Content/css/jquery.timepicker.css">

<link rel="stylesheet" href="~/Content/css/flaticon.css">

<link rel="stylesheet" href="~/Content/css/icomoon.css">

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark ftco\_navbar bg-dark ftco-navbar-light" id="ftco-navbar">

<div class="container">

<div class="navbar-brand"><img width="100" src="~/assets/images/images.jpg" /></div>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#ftco-nav" aria-controls="ftco-nav" aria-expanded="false" aria-label="Toggle navigation">

<span class="oi oi-menu"></span> Menu

</button>

<div class="collapse navbar-collapse" id="ftco-nav">

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

<li class="nav-item">@Html.ActionLink("Home", "Index", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("About", "About", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("Doctors", "Doctors", "Doctors", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("Contact", "Contact", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("QUERIES", "AllQuestions", "Questions", null, new { @class = "nav-link text-info font-weight-bold" })</li>

</ul>

<ul>

@Html.Partial("\_LoginPartial")

</ul>

</div>

</div>

</nav>

<div class="body-content">

@RenderBody()

<hr />

<footer class="ftco-footer ftco-bg-dark ftco-section img" style="background-image: url(../../assets/images/bg\_5.jpg);">

<div class="overlay"></div>

<div class="container">

<div class="row mb-5">

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Remedic</h2>

<p>Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts.</p>

<ul class="ftco-footer-social list-unstyled float-md-left float-lft mt-5">

<li class="ftco-animate"><a href="#"><span class="icon-twitter"></span></a></li>

<li class="ftco-animate"><a href="#"><span class="icon-facebook"></span></a></li>

<li class="ftco-animate"><a href="#"><span class="icon-instagram"></span></a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4 ml-md-5">

<h2 class="ftco-heading-2">Information</h2>

<ul class="list-unstyled">

<li><a href="#" class="py-2 d-block">Appointments</a></li>

<li><a href="#" class="py-2 d-block">Our Specialties</a></li>

<li><a href="#" class="py-2 d-block">Why Choose us</a></li>

<li><a href="#" class="py-2 d-block">Our Services</a></li>

<li><a href="#" class="py-2 d-block">health Tips</a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Site Links</h2>

<ul class="list-unstyled">

<li><a href="#" class="py-2 d-block">Home</a></li>

<li><a href="#" class="py-2 d-block">About</a></li>

<li><a href="#" class="py-2 d-block">Departments</a></li>

<li><a href="#" class="py-2 d-block">Doctors</a></li>

<li><a href="#" class="py-2 d-block">Blog</a></li>

<li><a href="#" class="py-2 d-block">Contact</a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Have a Questions?</h2>

<div class="block-23 mb-3">

<ul>

<li><span class="icon icon-map-marker"></span><span class="text">203 Fake St. Mountain View, San Francisco, California, USA</span></li>

<li><a href="#"><span class="icon icon-phone"></span><span class="text">+2 392 3929 210</span></a></li>

<li><a href="#"><span class="icon icon-envelope"></span><span class="text">info@yourdomain.com</span></a></li>

</ul>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12 text-center">

<p>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

Copyright &copy;

<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="icon-heart" aria-hidden="true"></i> by <a href="https://colorlib.com" target="\_blank">Colorlib</a>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

</p>

</div>

</div>

</div>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

<script src="~/Scripts/js/jquery.min.js"></script>

<script src="~/Scripts/js/jquery-migrate-3.0.1.min.js"></script>

<script src="~/Scripts/js/popper.min.js"></script>

<script src="~/Scripts/js/bootstrap.min.js"></script>

<script src="~/Scripts/js/jquery.easing.1.3.js"></script>

<script src="~/Scripts/js/jquery.waypoints.min.js"></script>

<script src="~/Scripts/js/jquery.stellar.min.js"></script>

<script src="~/Scripts/js/owl.carousel.min.js"></script>

<script src="~/Scripts/js/jquery.magnific-popup.min.js"></script>

<script src="~/Scripts/js/aos.js"></script>

<script src="~/Scripts/js/jquery.animateNumber.min.js"></script>

<script src="~/Scripts/js/bootstrap-datepicker.js"></script>

<script src="~/Scripts/js/jquery.timepicker.min.js"></script>

<script src="~/Scripts/js/scrollax.min.js"></script>

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7SQ74\_uJiULgl1JY0H2s&sensor=false"></script>

<script src="~/Scripts/js/google-map.js"></script>

<script src="~/Scripts/js/main.js"></script>

@RenderSection("scripts", required: false)

</body>

</html>

**\_LOGINPARTIAL.CSHTML**

<!DOCTYPE html>

<html lang="en">

<head>

<title>@ViewBag.Title - Doctors Forum</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,700" rel="stylesheet">

<link rel="stylesheet" href="~/Content/css/open-iconic-bootstrap.min.css">

<link rel="stylesheet" href="~/Content/css/animate.css">

<link rel="stylesheet" href="~/Content/css/owl.carousel.min.css">

<link rel="stylesheet" href="~/Content/css/owl.theme.default.min.css">

<link rel="stylesheet" href="~/Content/css/magnific-popup.css">

<link rel="stylesheet" href="~/Content/css/aos.css">

<link rel="stylesheet" href="~/Content/css/ionicons.min.css">

<link rel="stylesheet" href="~/Content/css/bootstrap-datepicker.css">

<link rel="stylesheet" href="~/Content/css/jquery.timepicker.css">

<link rel="stylesheet" href="~/Content/css/flaticon.css">

<link rel="stylesheet" href="~/Content/css/icomoon.css">

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark ftco\_navbar bg-dark ftco-navbar-light" id="ftco-navbar">

<div class="container">

<div class="navbar-brand"><img width="100" src="~/assets/images/images.jpg" /></div>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#ftco-nav" aria-controls="ftco-nav" aria-expanded="false" aria-label="Toggle navigation">

<span class="oi oi-menu"></span> Menu

</button>

<div class="collapse navbar-collapse" id="ftco-nav">

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

<li class="nav-item">@Html.ActionLink("Home", "Index", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("About", "About", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("Doctors", "Doctors", "Doctors", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("Contact", "Contact", "Home", null, new { @class = "nav-link" })</li>

<li class="nav-item">@Html.ActionLink("QUERIES", "AllQuestions", "Questions", null, new { @class = "nav-link text-info font-weight-bold" })</li>

</ul>

<ul>

@Html.Partial("\_LoginPartial")

</ul>

</div>

</div>

</nav>

<div class="body-content">

@RenderBody()

<hr />

<footer class="ftco-footer ftco-bg-dark ftco-section img" style="background-image: url(../../assets/images/bg\_5.jpg);">

<div class="overlay"></div>

<div class="container">

<div class="row mb-5">

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Remedic</h2>

<p>Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts.</p>

<ul class="ftco-footer-social list-unstyled float-md-left float-lft mt-5">

<li class="ftco-animate"><a href="#"><span class="icon-twitter"></span></a></li>

<li class="ftco-animate"><a href="#"><span class="icon-facebook"></span></a></li>

<li class="ftco-animate"><a href="#"><span class="icon-instagram"></span></a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4 ml-md-5">

<h2 class="ftco-heading-2">Information</h2>

<ul class="list-unstyled">

<li><a href="#" class="py-2 d-block">Appointments</a></li>

<li><a href="#" class="py-2 d-block">Our Specialties</a></li>

<li><a href="#" class="py-2 d-block">Why Choose us</a></li>

<li><a href="#" class="py-2 d-block">Our Services</a></li>

<li><a href="#" class="py-2 d-block">health Tips</a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Site Links</h2>

<ul class="list-unstyled">

<li><a href="#" class="py-2 d-block">Home</a></li>

<li><a href="#" class="py-2 d-block">About</a></li>

<li><a href="#" class="py-2 d-block">Departments</a></li>

<li><a href="#" class="py-2 d-block">Doctors</a></li>

<li><a href="#" class="py-2 d-block">Blog</a></li>

<li><a href="#" class="py-2 d-block">Contact</a></li>

</ul>

</div>

</div>

<div class="col-md">

<div class="ftco-footer-widget mb-4">

<h2 class="ftco-heading-2">Have a Questions?</h2>

<div class="block-23 mb-3">

<ul>

<li><span class="icon icon-map-marker"></span><span class="text">203 Fake St. Mountain View, San Francisco, California, USA</span></li>

<li><a href="#"><span class="icon icon-phone"></span><span class="text">+2 392 3929 210</span></a></li>

<li><a href="#"><span class="icon icon-envelope"></span><span class="text">info@yourdomain.com</span></a></li>

</ul>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12 text-center">

<p>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

Copyright &copy;

<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="icon-heart" aria-hidden="true"></i> by <a href="https://colorlib.com" target="\_blank">Colorlib</a>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

</p>

</div>

</div>

</div>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

<script src="~/Scripts/js/jquery.min.js"></script>

<script src="~/Scripts/js/jquery-migrate-3.0.1.min.js"></script>

<script src="~/Scripts/js/popper.min.js"></script>

<script src="~/Scripts/js/bootstrap.min.js"></script>

<script src="~/Scripts/js/jquery.easing.1.3.js"></script>

<script src="~/Scripts/js/jquery.waypoints.min.js"></script>

<script src="~/Scripts/js/jquery.stellar.min.js"></script>

<script src="~/Scripts/js/owl.carousel.min.js"></script>

<script src="~/Scripts/js/jquery.magnific-popup.min.js"></script>

<script src="~/Scripts/js/aos.js"></script>

<script src="~/Scripts/js/jquery.animateNumber.min.js"></script>

<script src="~/Scripts/js/bootstrap-datepicker.js"></script>

<script src="~/Scripts/js/jquery.timepicker.min.js"></script>

<script src="~/Scripts/js/scrollax.min.js"></script>

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7SQ74\_uJiULgl1JY0H2s&sensor=false"></script>

<script src="~/Scripts/js/google-map.js"></script>

<script src="~/Scripts/js/main.js"></script>

@RenderSection("scripts", required: false)

</body>

</html>

**ACCOUNT MODEL**

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

namespace SimpleChat.Models

{

public class ExternalLoginConfirmationViewModel

{

[Required]

[Display(Name = "Email")]

public string Email { get; set; }

}

public class ExternalLoginListViewModel

{

public string ReturnUrl { get; set; }

}

public class SendCodeViewModel

{

public string SelectedProvider { get; set; }

public ICollection<System.Web.Mvc.SelectListItem> Providers { get; set; }

public string ReturnUrl { get; set; }

public bool RememberMe { get; set; }

}

public class VerifyCodeViewModel

{

[Required]

public string Provider { get; set; }

[Required]

[Display(Name = "Code")]

public string Code { get; set; }

public string ReturnUrl { get; set; }

[Display(Name = "Remember this browser?")]

public bool RememberBrowser { get; set; }

public bool RememberMe { get; set; }

}

public class ForgotViewModel

{

[Required]

[Display(Name = "Email")]

public string Email { get; set; }

}

public class LoginViewModel

{

[Required]

[Display(Name = "Email")]

[EmailAddress]

public string Email { get; set; }

[Required]

[DataType(DataType.Password)]

[Display(Name = "Password")]

public string Password { get; set; }

[Display(Name = "Remember me?")]

public bool RememberMe { get; set; }

}

public class RegisterViewModel

{

[Required]

[Display(Name = "Full Name")]

public string FullName { get; set; }

[Required, Display(Name = "Phone Number")]

public string Telephone { get; set; }

[Required]

public string Address { get; set; }

[Required]

public string Profession { get; set; }

[Required, Display(Name = "Qualification(s)")]

public string Qualification { get; set; }

[Required, Display(Name = "Tell us your experience so far in your field")]

public string Experience { get; set; }

[Required, Display(Name = "Achievement(s)")]

public string Achievement { get; set; }

[Required]

[EmailAddress]

[Display(Name = "Email")]

public string Email { get; set; }

[Required]

[StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.", MinimumLength = 6)]

[DataType(DataType.Password)]

[Display(Name = "Password")]

public string Password { get; set; }

[DataType(DataType.Password)]

[Display(Name = "Confirm password")]

[Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]

public string ConfirmPassword { get; set; }

}

public class ResetPasswordViewModel

{

[Required]

[EmailAddress]

[Display(Name = "Email")]

public string Email { get; set; }

[Required]

[StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.", MinimumLength = 6)]

[DataType(DataType.Password)]

[Display(Name = "Password")]

public string Password { get; set; }

[DataType(DataType.Password)]

[Display(Name = "Confirm password")]

[Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]

public string ConfirmPassword { get; set; }

public string Code { get; set; }

}

public class ForgotPasswordViewModel

{

[Required]

[EmailAddress]

[Display(Name = "Email")]

public string Email { get; set; }

}

}

**QUESTIONS MODEL**

using System;

using System.Linq;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

using Microsoft.AspNet.Identity;

using Microsoft.AspNet.Identity.Owin;

using Microsoft.Owin.Security;

using SimpleChat.Models;

namespace SimpleChat.Controllers

{

[Authorize]

public class ManageController : Controller

{

private ApplicationSignInManager \_signInManager;

private ApplicationUserManager \_userManager;

public ManageController()

{

}

public ManageController(ApplicationUserManager userManager, ApplicationSignInManager signInManager)

{

UserManager = userManager;

SignInManager = signInManager;

}

public ApplicationSignInManager SignInManager

{

get

{

return \_signInManager ?? HttpContext.GetOwinContext().Get<ApplicationSignInManager>();

}

private set

{

\_signInManager = value;

}

}

public ApplicationUserManager UserManager

{

get

{

return \_userManager ?? HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();

}

private set

{

\_userManager = value;

}

}

//

// GET: /Manage/Index

public async Task<ActionResult> Index(ManageMessageId? message)

{

ViewBag.StatusMessage =

message == ManageMessageId.ChangePasswordSuccess ? "Your password has been changed."

: message == ManageMessageId.SetPasswordSuccess ? "Your password has been set."

: message == ManageMessageId.SetTwoFactorSuccess ? "Your two-factor authentication provider has been set."

: message == ManageMessageId.Error ? "An error has occurred."

: message == ManageMessageId.AddPhoneSuccess ? "Your phone number was added."

: message == ManageMessageId.RemovePhoneSuccess ? "Your phone number was removed."

: "";

var userId = User.Identity.GetUserId();

var model = new IndexViewModel

{

HasPassword = HasPassword(),

PhoneNumber = await UserManager.GetPhoneNumberAsync(userId),

TwoFactor = await UserManager.GetTwoFactorEnabledAsync(userId),

Logins = await UserManager.GetLoginsAsync(userId),

BrowserRemembered = await AuthenticationManager.TwoFactorBrowserRememberedAsync(userId)

};

return View(model);

}

//

// POST: /Manage/RemoveLogin

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> RemoveLogin(string loginProvider, string providerKey)

{

ManageMessageId? message;

var result = await UserManager.RemoveLoginAsync(User.Identity.GetUserId(), new UserLoginInfo(loginProvider, providerKey));

if (result.Succeeded)

{

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

message = ManageMessageId.RemoveLoginSuccess;

}

else

{

message = ManageMessageId.Error;

}

return RedirectToAction("ManageLogins", new { Message = message });

}

//

// GET: /Manage/AddPhoneNumber

public ActionResult AddPhoneNumber()

{

return View();

}

//

// POST: /Manage/AddPhoneNumber

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> AddPhoneNumber(AddPhoneNumberViewModel model)

{

if (!ModelState.IsValid)

{

return View(model);

}

// Generate the token and send it

var code = await UserManager.GenerateChangePhoneNumberTokenAsync(User.Identity.GetUserId(), model.Number);

if (UserManager.SmsService != null)

{

var message = new IdentityMessage

{

Destination = model.Number,

Body = "Your security code is: " + code

};

await UserManager.SmsService.SendAsync(message);

}

return RedirectToAction("VerifyPhoneNumber", new { PhoneNumber = model.Number });

}

//

// POST: /Manage/EnableTwoFactorAuthentication

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> EnableTwoFactorAuthentication()

{

await UserManager.SetTwoFactorEnabledAsync(User.Identity.GetUserId(), true);

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", "Manage");

}

//

// POST: /Manage/DisableTwoFactorAuthentication

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> DisableTwoFactorAuthentication()

{

await UserManager.SetTwoFactorEnabledAsync(User.Identity.GetUserId(), false);

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", "Manage");

}

//

// GET: /Manage/VerifyPhoneNumber

public async Task<ActionResult> VerifyPhoneNumber(string phoneNumber)

{

var code = await UserManager.GenerateChangePhoneNumberTokenAsync(User.Identity.GetUserId(), phoneNumber);

// Send an SMS through the SMS provider to verify the phone number

return phoneNumber == null ? View("Error") : View(new VerifyPhoneNumberViewModel { PhoneNumber = phoneNumber });

}

//

// POST: /Manage/VerifyPhoneNumber

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> VerifyPhoneNumber(VerifyPhoneNumberViewModel model)

{

if (!ModelState.IsValid)

{

return View(model);

}

var result = await UserManager.ChangePhoneNumberAsync(User.Identity.GetUserId(), model.PhoneNumber, model.Code);

if (result.Succeeded)

{

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", new { Message = ManageMessageId.AddPhoneSuccess });

}

// If we got this far, something failed, redisplay form

ModelState.AddModelError("", "Failed to verify phone");

return View(model);

}

//

// POST: /Manage/RemovePhoneNumber

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> RemovePhoneNumber()

{

var result = await UserManager.SetPhoneNumberAsync(User.Identity.GetUserId(), null);

if (!result.Succeeded)

{

return RedirectToAction("Index", new { Message = ManageMessageId.Error });

}

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", new { Message = ManageMessageId.RemovePhoneSuccess });

}

//

// GET: /Manage/ChangePassword

public ActionResult ChangePassword()

{

return View();

}

//

// POST: /Manage/ChangePassword

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> ChangePassword(ChangePasswordViewModel model)

{

if (!ModelState.IsValid)

{

return View(model);

}

var result = await UserManager.ChangePasswordAsync(User.Identity.GetUserId(), model.OldPassword, model.NewPassword);

if (result.Succeeded)

{

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", new { Message = ManageMessageId.ChangePasswordSuccess });

}

AddErrors(result);

return View(model);

}

//

// GET: /Manage/SetPassword

public ActionResult SetPassword()

{

return View();

}

//

// POST: /Manage/SetPassword

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> SetPassword(SetPasswordViewModel model)

{

if (ModelState.IsValid)

{

var result = await UserManager.AddPasswordAsync(User.Identity.GetUserId(), model.NewPassword);

if (result.Succeeded)

{

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user != null)

{

await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);

}

return RedirectToAction("Index", new { Message = ManageMessageId.SetPasswordSuccess });

}

AddErrors(result);

}

// If we got this far, something failed, redisplay form

return View(model);

}

//

// GET: /Manage/ManageLogins

public async Task<ActionResult> ManageLogins(ManageMessageId? message)

{

ViewBag.StatusMessage =

message == ManageMessageId.RemoveLoginSuccess ? "The external login was removed."

: message == ManageMessageId.Error ? "An error has occurred."

: "";

var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

if (user == null)

{

return View("Error");

}

var userLogins = await UserManager.GetLoginsAsync(User.Identity.GetUserId());

var otherLogins = AuthenticationManager.GetExternalAuthenticationTypes().Where(auth => userLogins.All(ul => auth.AuthenticationType != ul.LoginProvider)).ToList();

ViewBag.ShowRemoveButton = user.PasswordHash != null || userLogins.Count > 1;

return View(new ManageLoginsViewModel

{

CurrentLogins = userLogins,

OtherLogins = otherLogins

});

}

//

// POST: /Manage/LinkLogin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult LinkLogin(string provider)

{

// Request a redirect to the external login provider to link a login for the current user

return new AccountController.ChallengeResult(provider, Url.Action("LinkLoginCallback", "Manage"), User.Identity.GetUserId());

}

//

// GET: /Manage/LinkLoginCallback

public async Task<ActionResult> LinkLoginCallback()

{

var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync(XsrfKey, User.Identity.GetUserId());

if (loginInfo == null)

{

return RedirectToAction("ManageLogins", new { Message = ManageMessageId.Error });

}

var result = await UserManager.AddLoginAsync(User.Identity.GetUserId(), loginInfo.Login);

return result.Succeeded ? RedirectToAction("ManageLogins") : RedirectToAction("ManageLogins", new { Message = ManageMessageId.Error });

}

protected override void Dispose(bool disposing)

{

if (disposing && \_userManager != null)

{

\_userManager.Dispose();

\_userManager = null;

}

base.Dispose(disposing);

}

#region Helpers

// Used for XSRF protection when adding external logins

private const string XsrfKey = "XsrfId";

private IAuthenticationManager AuthenticationManager

{

get

{

return HttpContext.GetOwinContext().Authentication;

}

}

private void AddErrors(IdentityResult result)

{

foreach (var error in result.Errors)

{

ModelState.AddModelError("", error);

}

}

private bool HasPassword()

{

var user = UserManager.FindById(User.Identity.GetUserId());

if (user != null)

{

return user.PasswordHash != null;

}

return false;

}

private bool HasPhoneNumber()

{

var user = UserManager.FindById(User.Identity.GetUserId());

if (user != null)

{

return user.PhoneNumber != null;

}

return false;

}

public enum ManageMessageId

{

AddPhoneSuccess,

ChangePasswordSuccess,

SetTwoFactorSuccess,

SetPasswordSuccess,

RemoveLoginSuccess,

RemovePhoneSuccess,

Error

}

#endregion

}

}

**REPLIES MODEL**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace SimpleChat.Models

{

public class Replies

{

public int Id { get; set; }

[Display(Name = "Your Reply")]

public string RepliesText { get; set; }

public Questions Question { get; set; }

public int QuestionId { get; set; }

public ApplicationUser User { get; set; }

public string UserId { get; set; }

public DateTime ReplyDate { get; set; }

}

}

**DOCTORS CONTROLLER**

using SimpleChat.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace SimpleChat.Controllers

{

public class DoctorsController : Controller

{

private ApplicationDbContext \_db;

//This is the public constructor for the Doctors Controller. the Application database context was Initialized here.

public DoctorsController()

{

\_db = new ApplicationDbContext();

}

//it is a protected function that was overriden from the controller class. it is used to close the Application database context opened in the constructor to avoid unhandle datbase openings.

protected override void Dispose(bool disposing)

{

\_db.Dispose();

}

//it is a public function that returns a View with a list of all the doctors registered in this application.

// GET: Doctors

public ActionResult Doctors()

{

ViewBag.userCount = \_db.Users.Count();

var Users = \_db.Users.ToList();

return View(Users);

}

}

}

**QUESTIONS CONTROLLER**

using Microsoft.AspNet.Identity;

using SimpleChat.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Data.Entity;

namespace SimpleChat.Controllers

{

public class QuestionsController : Controller

{

private ApplicationDbContext \_db;

//This is the public constructor for the Questions Controller. the Application database context was Initialized here

public QuestionsController()

{

\_db = new ApplicationDbContext();

}

//it is a protected function that was overriden from the controller class. it is used to close the Application database context opened in the constructor

protected override void Dispose(bool disposing)

{

\_db.Dispose();

}

//this is a public function that returns the view for the questions Page

[Authorize]

// GET: Questions

public ActionResult Index()

{

return View();

}

//This is a public function that adds a new question submitted by a user in the application. it takes an Object parameter of Question Model Class then saves it in the Database

public ActionResult Add(Questions questions)

{

questions.DateSent = DateTime.Now;

string user = User.Identity.GetUserId();

questions.UserId = user;

\_db.Questions.Add(questions);

\_db.SaveChanges();

return RedirectToAction("AllQuestions");

}

//it is a public function that returns a View with a list of all the questions asked by users in this application.

public ActionResult AllQuestions()

{

ViewBag.userCount = \_db.Users.Count();

var questions = \_db.Questions.Include(u => u.User).ToList();

return View(questions);

}

}

}

**REPLIES CONTROLLER**

using Microsoft.AspNet.Identity;

using SimpleChat.chatModel;

using SimpleChat.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Data.Entity;

namespace SimpleChat.Controllers

{

public class RepliesController : Controller

{

private ApplicationDbContext \_db;

//This is the public constructor for the Replies Controller. the Application database context was Initialized here.

public RepliesController()

{

\_db = new ApplicationDbContext();

}

//it is a protected function that was overriden from the controller class. it is used to close the Application database context opened in the constructor.

protected override void Dispose(bool disposing)

{

\_db.Dispose();

}

//it is a public function that returns a View with a list of all the replies for a particular question asked by a user in this application

// GET: Replies

public ActionResult Index()

{

return View();

}

//It is a public function with a parameter Id of type Integer used to get the question then returns a view with the question gotten by the parameter passed and with a list of all the replies for the question

[Authorize]

public ActionResult Reply(int id)

{

var ViewAllReply = \_db.Replies.Include(u => u.User).Where(c => c.QuestionId == id).ToList();

var viewSpecificQuestion = \_db.Questions.Where(c => c.Id == id).ToList();

var view = new ViewReplies

{

Reply = ViewAllReply,

Questions = viewSpecificQuestion

};

return View(view);

}

//It is a public function with a parameter of type Object of the Replies Model Class and id of the question. it is used to save replies for a particular Question and returns a view with the question and a list of replies for the question

public ActionResult ReplySave(Replies replies, int id)

{

string user = User.Identity.GetUserId();

replies.QuestionId = id;

replies.ReplyDate = DateTime.Now;

replies.UserId = user;

\_db.Replies.Add(replies);

\_db.SaveChanges();

// Replies to specific question

var AllReplies = \_db.Replies.Include(u => u.User).Where(c => c.QuestionId == id).ToList();

var RepliedQuestion = \_db.Questions.Where(c => c.Id == id).ToList();

// Using View Model, Joining two tables together

var ViewModel = new ViewReplies

{

Reply = AllReplies,

Questions = RepliedQuestion

};

return View(ViewModel);

}

//It is a public function with a parameter Id of type Integer used to get all the replies for a particular Question.It returns a view with the list of replies for the question

public ActionResult ViewAll(int id)

{

var ViewAllReply = \_db.Replies.Include(u => u.User).Where(c => c.QuestionId == id).ToList();

var viewSpecificQuestion = \_db.Questions.Where(c => c.Id == id).ToList();

var view = new ViewReplies

{

Reply = ViewAllReply,

Questions = viewSpecificQuestion

};

return View(view);

}

}

}

**ACCOUNT CONTROLLER**

**using System;**

**using System.Globalization;**

**using System.Linq;**

**using System.Security.Claims;**

**using System.Threading.Tasks;**

**using System.Web;**

**using System.Web.Mvc;**

**using Microsoft.AspNet.Identity;**

**using Microsoft.AspNet.Identity.Owin;**

**using Microsoft.Owin.Security;**

**using SimpleChat.Models;**

**namespace SimpleChat.Controllers**

**{**

**[Authorize]**

**public class AccountController : Controller**

**{**

**private ApplicationSignInManager \_signInManager;**

**private ApplicationUserManager \_userManager;**

**public AccountController()**

**{**

**}**

**public AccountController(ApplicationUserManager userManager, ApplicationSignInManager signInManager )**

**{**

**UserManager = userManager;**

**SignInManager = signInManager;**

**}**

**public ApplicationSignInManager SignInManager**

**{**

**get**

**{**

**return \_signInManager ?? HttpContext.GetOwinContext().Get<ApplicationSignInManager>();**

**}**

**private set**

**{**

**\_signInManager = value;**

**}**

**}**

**public ApplicationUserManager UserManager**

**{**

**get**

**{**

**return \_userManager ?? HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();**

**}**

**private set**

**{**

**\_userManager = value;**

**}**

**}**

**//**

**// GET: /Account/Login**

**[AllowAnonymous]**

**public ActionResult Login(string returnUrl)**

**{**

**ViewBag.ReturnUrl = returnUrl;**

**return View();**

**}**

**//**

**// POST: /Account/Login**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**// This doesn't count login failures towards account lockout**

**// To enable password failures to trigger account lockout, change to shouldLockout: true**

**var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, shouldLockout: false);**

**switch (result)**

**{**

**case SignInStatus.Success:**

**return RedirectToLocal(returnUrl);**

**case SignInStatus.LockedOut:**

**return View("Lockout");**

**case SignInStatus.RequiresVerification:**

**return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });**

**case SignInStatus.Failure:**

**default:**

**ModelState.AddModelError("", "Invalid login attempt.");**

**return View(model);**

**}**

**}**

**//**

**// GET: /Account/VerifyCode**

**[AllowAnonymous]**

**public async Task<ActionResult> VerifyCode(string provider, string returnUrl, bool rememberMe)**

**{**

**// Require that the user has already logged in via username/password or external login**

**if (!await SignInManager.HasBeenVerifiedAsync())**

**{**

**return View("Error");**

**}**

**return View(new VerifyCodeViewModel { Provider = provider, ReturnUrl = returnUrl, RememberMe = rememberMe });**

**}**

**//**

**// POST: /Account/VerifyCode**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> VerifyCode(VerifyCodeViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**// The following code protects for brute force attacks against the two factor codes.**

**// If a user enters incorrect codes for a specified amount of time then the user account**

**// will be locked out for a specified amount of time.**

**// You can configure the account lockout settings in IdentityConfig**

**var result = await SignInManager.TwoFactorSignInAsync(model.Provider, model.Code, isPersistent: model.RememberMe, rememberBrowser: model.RememberBrowser);**

**switch (result)**

**{**

**case SignInStatus.Success:**

**return RedirectToLocal(model.ReturnUrl);**

**case SignInStatus.LockedOut:**

**return View("Lockout");**

**case SignInStatus.Failure:**

**default:**

**ModelState.AddModelError("", "Invalid code.");**

**return View(model);**

**}**

**}**

**//**

**// GET: /Account/Register**

**[AllowAnonymous]**

**public ActionResult Register()**

**{**

**return View();**

**}**

**//**

**// POST: /Account/Register**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> Register(RegisterViewModel model)**

**{**

**if (ModelState.IsValid)**

**{**

**var user = new ApplicationUser {**

**UserName = model.Email,**

**Email = model.Email,**

**FullName = model.FullName,**

**Telephone = model.Telephone,**

**Address = model.Address,**

**Profession = model.Profession,**

**Qualification = model.Qualification,**

**Achievement = model.Achievement,**

**Experience = model.Experience**

**};**

**var result = await UserManager.CreateAsync(user, model.Password);**

**if (result.Succeeded)**

**{**

**await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);**

**// For more information on how to enable account confirmation and password reset please visit** [**https://go.microsoft.com/fwlink/?LinkID=320771**](https://go.microsoft.com/fwlink/?LinkID=320771)

**// Send an email with this link**

**// string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);**

**// var callbackUrl = Url.Action("ConfirmEmail", "Account", new { userId = user.Id, code = code }, protocol: Request.Url.Scheme);**

**// await UserManager.SendEmailAsync(user.Id, "Confirm your account", "Please confirm your account by clicking <a href=\"" + callbackUrl + "\">here</a>");**

**ViewBag.successMessage = "You've been registered successfully";**

**return View();**

**//return RedirectToAction("Index", "Home");**

**}**

**AddErrors(result);**

**}**

**// If we got this far, something failed, redisplay form**

**return View(model);**

**}**

**//**

**// GET: /Account/ConfirmEmail**

**[AllowAnonymous]**

**public async Task<ActionResult> ConfirmEmail(string userId, string code)**

**{**

**if (userId == null || code == null)**

**{**

**return View("Error");**

**}**

**var result = await UserManager.ConfirmEmailAsync(userId, code);**

**return View(result.Succeeded ? "ConfirmEmail" : "Error");**

**}**

**//**

**// GET: /Account/ForgotPassword**

**[AllowAnonymous]**

**public ActionResult ForgotPassword()**

**{**

**return View();**

**}**

**//**

**// POST: /Account/ForgotPassword**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> ForgotPassword(ForgotPasswordViewModel model)**

**{**

**if (ModelState.IsValid)**

**{**

**var user = await UserManager.FindByNameAsync(model.Email);**

**if (user == null || !(await UserManager.IsEmailConfirmedAsync(user.Id)))**

**{**

**// Don't reveal that the user does not exist or is not confirmed**

**return View("ForgotPasswordConfirmation");**

**}**

**// For more information on how to enable account confirmation and password reset please visit** [**https://go.microsoft.com/fwlink/?LinkID=320771**](https://go.microsoft.com/fwlink/?LinkID=320771)

**// Send an email with this link**

**// string code = await UserManager.GeneratePasswordResetTokenAsync(user.Id);**

**// var callbackUrl = Url.Action("ResetPassword", "Account", new { userId = user.Id, code = code }, protocol: Request.Url.Scheme);**

**// await UserManager.SendEmailAsync(user.Id, "Reset Password", "Please reset your password by clicking <a href=\"" + callbackUrl + "\">here</a>");**

**// return RedirectToAction("ForgotPasswordConfirmation", "Account");**

**}**

**// If we got this far, something failed, redisplay form**

**return View(model);**

**}**

**//**

**// GET: /Account/ForgotPasswordConfirmation**

**[AllowAnonymous]**

**public ActionResult ForgotPasswordConfirmation()**

**{**

**return View();**

**}**

**//**

**// GET: /Account/ResetPassword**

**[AllowAnonymous]**

**public ActionResult ResetPassword(string code)**

**{**

**return code == null ? View("Error") : View();**

**}**

**//**

**// POST: /Account/ResetPassword**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> ResetPassword(ResetPasswordViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**var user = await UserManager.FindByNameAsync(model.Email);**

**if (user == null)**

**{**

**// Don't reveal that the user does not exist**

**return RedirectToAction("ResetPasswordConfirmation", "Account");**

**}**

**var result = await UserManager.ResetPasswordAsync(user.Id, model.Code, model.Password);**

**if (result.Succeeded)**

**{**

**return RedirectToAction("ResetPasswordConfirmation", "Account");**

**}**

**AddErrors(result);**

**return View();**

**}**

**//**

**// GET: /Account/ResetPasswordConfirmation**

**[AllowAnonymous]**

**public ActionResult ResetPasswordConfirmation()**

**{**

**return View();**

**}**

**//**

**// POST: /Account/ExternalLogin**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public ActionResult ExternalLogin(string provider, string returnUrl)**

**{**

**// Request a redirect to the external login provider**

**return new ChallengeResult(provider, Url.Action("ExternalLoginCallback", "Account", new { ReturnUrl = returnUrl }));**

**}**

**//**

**// GET: /Account/SendCode**

**[AllowAnonymous]**

**public async Task<ActionResult> SendCode(string returnUrl, bool rememberMe)**

**{**

**var userId = await SignInManager.GetVerifiedUserIdAsync();**

**if (userId == null)**

**{**

**return View("Error");**

**}**

**var userFactors = await UserManager.GetValidTwoFactorProvidersAsync(userId);**

**var factorOptions = userFactors.Select(purpose => new SelectListItem { Text = purpose, Value = purpose }).ToList();**

**return View(new SendCodeViewModel { Providers = factorOptions, ReturnUrl = returnUrl, RememberMe = rememberMe });**

**}**

**//**

**// POST: /Account/SendCode**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> SendCode(SendCodeViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View();**

**}**

**// Generate the token and send it**

**if (!await SignInManager.SendTwoFactorCodeAsync(model.SelectedProvider))**

**{**

**return View("Error");**

**}**

**return RedirectToAction("VerifyCode", new { Provider = model.SelectedProvider, ReturnUrl = model.ReturnUrl, RememberMe = model.RememberMe });**

**}**

**//**

**// GET: /Account/ExternalLoginCallback**

**[AllowAnonymous]**

**public async Task<ActionResult> ExternalLoginCallback(string returnUrl)**

**{**

**var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync();**

**if (loginInfo == null)**

**{**

**return RedirectToAction("Login");**

**}**

**// Sign in the user with this external login provider if the user already has a login**

**var result = await SignInManager.ExternalSignInAsync(loginInfo, isPersistent: false);**

**switch (result)**

**{**

**case SignInStatus.Success:**

**return RedirectToLocal(returnUrl);**

**case SignInStatus.LockedOut:**

**return View("Lockout");**

**case SignInStatus.RequiresVerification:**

**return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = false });**

**case SignInStatus.Failure:**

**default:**

**// If the user does not have an account, then prompt the user to create an account**

**ViewBag.ReturnUrl = returnUrl;**

**ViewBag.LoginProvider = loginInfo.Login.LoginProvider;**

**return View("ExternalLoginConfirmation", new ExternalLoginConfirmationViewModel { Email = loginInfo.Email });**

**}**

**}**

**//**

**// POST: /Account/ExternalLoginConfirmation**

**[HttpPost]**

**[AllowAnonymous]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> ExternalLoginConfirmation(ExternalLoginConfirmationViewModel model, string returnUrl)**

**{**

**if (User.Identity.IsAuthenticated)**

**{**

**return RedirectToAction("Index", "Manage");**

**}**

**if (ModelState.IsValid)**

**{**

**// Get the information about the user from the external login provider**

**var info = await AuthenticationManager.GetExternalLoginInfoAsync();**

**if (info == null)**

**{**

**return View("ExternalLoginFailure");**

**}**

**var user = new ApplicationUser { UserName = model.Email, Email = model.Email };**

**var result = await UserManager.CreateAsync(user);**

**if (result.Succeeded)**

**{**

**result = await UserManager.AddLoginAsync(user.Id, info.Login);**

**if (result.Succeeded)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**return RedirectToLocal(returnUrl);**

**}**

**}**

**AddErrors(result);**

**}**

**ViewBag.ReturnUrl = returnUrl;**

**return View(model);**

**}**

**//**

**// POST: /Account/LogOff**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public ActionResult LogOff()**

**{**

**AuthenticationManager.SignOut(DefaultAuthenticationTypes.ApplicationCookie);**

**return RedirectToAction("Index", "Home");**

**}**

**//**

**// GET: /Account/ExternalLoginFailure**

**[AllowAnonymous]**

**public ActionResult ExternalLoginFailure()**

**{**

**return View();**

**}**

**protected override void Dispose(bool disposing)**

**{**

**if (disposing)**

**{**

**if (\_userManager != null)**

**{**

**\_userManager.Dispose();**

**\_userManager = null;**

**}**

**if (\_signInManager != null)**

**{**

**\_signInManager.Dispose();**

**\_signInManager = null;**

**}**

**}**

**base.Dispose(disposing);**

**}**

**#region Helpers**

**// Used for XSRF protection when adding external logins**

**private const string XsrfKey = "XsrfId";**

**private IAuthenticationManager AuthenticationManager**

**{**

**get**

**{**

**return HttpContext.GetOwinContext().Authentication;**

**}**

**}**

**private void AddErrors(IdentityResult result)**

**{**

**foreach (var error in result.Errors)**

**{**

**ModelState.AddModelError("", error);**

**}**

**}**

**private ActionResult RedirectToLocal(string returnUrl)**

**{**

**if (Url.IsLocalUrl(returnUrl))**

**{**

**return Redirect(returnUrl);**

**}**

**return RedirectToAction("Index", "Home");**

**}**

**internal class ChallengeResult : HttpUnauthorizedResult**

**{**

**public ChallengeResult(string provider, string redirectUri)**

**: this(provider, redirectUri, null)**

**{**

**}**

**public ChallengeResult(string provider, string redirectUri, string userId)**

**{**

**LoginProvider = provider;**

**RedirectUri = redirectUri;**

**UserId = userId;**

**}**

**public string LoginProvider { get; set; }**

**public string RedirectUri { get; set; }**

**public string UserId { get; set; }**

**public override void ExecuteResult(ControllerContext context)**

**{**

**var properties = new AuthenticationProperties { RedirectUri = RedirectUri };**

**if (UserId != null)**

**{**

**properties.Dictionary[XsrfKey] = UserId;**

**}**

**context.HttpContext.GetOwinContext().Authentication.Challenge(properties, LoginProvider);**

**}**

**}**

**#endregion**

**}**

**}**

**HOME CONTROLLER**

using SimpleChat.chatModel;

using SimpleChat.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace SimpleChat.Controllers

{

public class HomeController : Controller

{

private ApplicationDbContext \_db;

public HomeController()

{

\_db = new ApplicationDbContext();

ViewBag.UserCount = \_db.Users.Count();

}

protected override void Dispose(bool disposing)

{

\_db.Dispose();

}

public ActionResult Index()

{

ViewBag.SuccessMessage = "You've been registered Successfully";

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

**MANAGE CONTROLLER**

**using System;**

**using System.Linq;**

**using System.Threading.Tasks;**

**using System.Web;**

**using System.Web.Mvc;**

**using Microsoft.AspNet.Identity;**

**using Microsoft.AspNet.Identity.Owin;**

**using Microsoft.Owin.Security;**

**using SimpleChat.Models;**

**namespace SimpleChat.Controllers**

**{**

**[Authorize]**

**public class ManageController : Controller**

**{**

**private ApplicationSignInManager \_signInManager;**

**private ApplicationUserManager \_userManager;**

**public ManageController()**

**{**

**}**

**public ManageController(ApplicationUserManager userManager, ApplicationSignInManager signInManager)**

**{**

**UserManager = userManager;**

**SignInManager = signInManager;**

**}**

**public ApplicationSignInManager SignInManager**

**{**

**get**

**{**

**return \_signInManager ?? HttpContext.GetOwinContext().Get<ApplicationSignInManager>();**

**}**

**private set**

**{**

**\_signInManager = value;**

**}**

**}**

**public ApplicationUserManager UserManager**

**{**

**get**

**{**

**return \_userManager ?? HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();**

**}**

**private set**

**{**

**\_userManager = value;**

**}**

**}**

**//**

**// GET: /Manage/Index**

**public async Task<ActionResult> Index(ManageMessageId? message)**

**{**

**ViewBag.StatusMessage =**

**message == ManageMessageId.ChangePasswordSuccess ? "Your password has been changed."**

**: message == ManageMessageId.SetPasswordSuccess ? "Your password has been set."**

**: message == ManageMessageId.SetTwoFactorSuccess ? "Your two-factor authentication provider has been set."**

**: message == ManageMessageId.Error ? "An error has occurred."**

**: message == ManageMessageId.AddPhoneSuccess ? "Your phone number was added."**

**: message == ManageMessageId.RemovePhoneSuccess ? "Your phone number was removed."**

**: "";**

**var userId = User.Identity.GetUserId();**

**var model = new IndexViewModel**

**{**

**HasPassword = HasPassword(),**

**PhoneNumber = await UserManager.GetPhoneNumberAsync(userId),**

**TwoFactor = await UserManager.GetTwoFactorEnabledAsync(userId),**

**Logins = await UserManager.GetLoginsAsync(userId),**

**BrowserRemembered = await AuthenticationManager.TwoFactorBrowserRememberedAsync(userId)**

**};**

**return View(model);**

**}**

**//**

**// POST: /Manage/RemoveLogin**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> RemoveLogin(string loginProvider, string providerKey)**

**{**

**ManageMessageId? message;**

**var result = await UserManager.RemoveLoginAsync(User.Identity.GetUserId(), new UserLoginInfo(loginProvider, providerKey));**

**if (result.Succeeded)**

**{**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**message = ManageMessageId.RemoveLoginSuccess;**

**}**

**else**

**{**

**message = ManageMessageId.Error;**

**}**

**return RedirectToAction("ManageLogins", new { Message = message });**

**}**

**//**

**// GET: /Manage/AddPhoneNumber**

**public ActionResult AddPhoneNumber()**

**{**

**return View();**

**}**

**//**

**// POST: /Manage/AddPhoneNumber**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> AddPhoneNumber(AddPhoneNumberViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**// Generate the token and send it**

**var code = await UserManager.GenerateChangePhoneNumberTokenAsync(User.Identity.GetUserId(), model.Number);**

**if (UserManager.SmsService != null)**

**{**

**var message = new IdentityMessage**

**{**

**Destination = model.Number,**

**Body = "Your security code is: " + code**

**};**

**await UserManager.SmsService.SendAsync(message);**

**}**

**return RedirectToAction("VerifyPhoneNumber", new { PhoneNumber = model.Number });**

**}**

**//**

**// POST: /Manage/EnableTwoFactorAuthentication**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> EnableTwoFactorAuthentication()**

**{**

**await UserManager.SetTwoFactorEnabledAsync(User.Identity.GetUserId(), true);**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", "Manage");**

**}**

**//**

**// POST: /Manage/DisableTwoFactorAuthentication**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> DisableTwoFactorAuthentication()**

**{**

**await UserManager.SetTwoFactorEnabledAsync(User.Identity.GetUserId(), false);**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", "Manage");**

**}**

**//**

**// GET: /Manage/VerifyPhoneNumber**

**public async Task<ActionResult> VerifyPhoneNumber(string phoneNumber)**

**{**

**var code = await UserManager.GenerateChangePhoneNumberTokenAsync(User.Identity.GetUserId(), phoneNumber);**

**// Send an SMS through the SMS provider to verify the phone number**

**return phoneNumber == null ? View("Error") : View(new VerifyPhoneNumberViewModel { PhoneNumber = phoneNumber });**

**}**

**//**

**// POST: /Manage/VerifyPhoneNumber**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> VerifyPhoneNumber(VerifyPhoneNumberViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**var result = await UserManager.ChangePhoneNumberAsync(User.Identity.GetUserId(), model.PhoneNumber, model.Code);**

**if (result.Succeeded)**

**{**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", new { Message = ManageMessageId.AddPhoneSuccess });**

**}**

**// If we got this far, something failed, redisplay form**

**ModelState.AddModelError("", "Failed to verify phone");**

**return View(model);**

**}**

**//**

**// POST: /Manage/RemovePhoneNumber**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> RemovePhoneNumber()**

**{**

**var result = await UserManager.SetPhoneNumberAsync(User.Identity.GetUserId(), null);**

**if (!result.Succeeded)**

**{**

**return RedirectToAction("Index", new { Message = ManageMessageId.Error });**

**}**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", new { Message = ManageMessageId.RemovePhoneSuccess });**

**}**

**//**

**// GET: /Manage/ChangePassword**

**public ActionResult ChangePassword()**

**{**

**return View();**

**}**

**//**

**// POST: /Manage/ChangePassword**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> ChangePassword(ChangePasswordViewModel model)**

**{**

**if (!ModelState.IsValid)**

**{**

**return View(model);**

**}**

**var result = await UserManager.ChangePasswordAsync(User.Identity.GetUserId(), model.OldPassword, model.NewPassword);**

**if (result.Succeeded)**

**{**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", new { Message = ManageMessageId.ChangePasswordSuccess });**

**}**

**AddErrors(result);**

**return View(model);**

**}**

**//**

**// GET: /Manage/SetPassword**

**public ActionResult SetPassword()**

**{**

**return View();**

**}**

**//**

**// POST: /Manage/SetPassword**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<ActionResult> SetPassword(SetPasswordViewModel model)**

**{**

**if (ModelState.IsValid)**

**{**

**var result = await UserManager.AddPasswordAsync(User.Identity.GetUserId(), model.NewPassword);**

**if (result.Succeeded)**

**{**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user != null)**

**{**

**await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);**

**}**

**return RedirectToAction("Index", new { Message = ManageMessageId.SetPasswordSuccess });**

**}**

**AddErrors(result);**

**}**

**// If we got this far, something failed, redisplay form**

**return View(model);**

**}**

**//**

**// GET: /Manage/ManageLogins**

**public async Task<ActionResult> ManageLogins(ManageMessageId? message)**

**{**

**ViewBag.StatusMessage =**

**message == ManageMessageId.RemoveLoginSuccess ? "The external login was removed."**

**: message == ManageMessageId.Error ? "An error has occurred."**

**: "";**

**var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());**

**if (user == null)**

**{**

**return View("Error");**

**}**

**var userLogins = await UserManager.GetLoginsAsync(User.Identity.GetUserId());**

**var otherLogins = AuthenticationManager.GetExternalAuthenticationTypes().Where(auth => userLogins.All(ul => auth.AuthenticationType != ul.LoginProvider)).ToList();**

**ViewBag.ShowRemoveButton = user.PasswordHash != null || userLogins.Count > 1;**

**return View(new ManageLoginsViewModel**

**{**

**CurrentLogins = userLogins,**

**OtherLogins = otherLogins**

**});**

**}**

**//**

**// POST: /Manage/LinkLogin**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public ActionResult LinkLogin(string provider)**

**{**

**// Request a redirect to the external login provider to link a login for the current user**

**return new AccountController.ChallengeResult(provider, Url.Action("LinkLoginCallback", "Manage"), User.Identity.GetUserId());**

**}**

**//**

**// GET: /Manage/LinkLoginCallback**

**public async Task<ActionResult> LinkLoginCallback()**

**{**

**var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync(XsrfKey, User.Identity.GetUserId());**

**if (loginInfo == null)**

**{**

**return RedirectToAction("ManageLogins", new { Message = ManageMessageId.Error });**

**}**

**var result = await UserManager.AddLoginAsync(User.Identity.GetUserId(), loginInfo.Login);**

**return result.Succeeded ? RedirectToAction("ManageLogins") : RedirectToAction("ManageLogins", new { Message = ManageMessageId.Error });**

**}**

**protected override void Dispose(bool disposing)**

**{**

**if (disposing && \_userManager != null)**

**{**

**\_userManager.Dispose();**

**\_userManager = null;**

**}**

**base.Dispose(disposing);**

**}**

**#region Helpers**

**// Used for XSRF protection when adding external logins**

**private const string XsrfKey = "XsrfId";**

**private IAuthenticationManager AuthenticationManager**

**{**

**get**

**{**

**return HttpContext.GetOwinContext().Authentication;**

**}**

**}**

**private void AddErrors(IdentityResult result)**

**{**

**foreach (var error in result.Errors)**

**{**

**ModelState.AddModelError("", error);**

**}**

**}**

**private bool HasPassword()**

**{**

**var user = UserManager.FindById(User.Identity.GetUserId());**

**if (user != null)**

**{**

**return user.PasswordHash != null;**

**}**

**return false;**

**}**

**private bool HasPhoneNumber()**

**{**

**var user = UserManager.FindById(User.Identity.GetUserId());**

**if (user != null)**

**{**

**return user.PhoneNumber != null;**

**}**

**return false;**

**}**

**public enum ManageMessageId**

**{**

**AddPhoneSuccess,**

**ChangePasswordSuccess,**

**SetTwoFactorSuccess,**

**SetPasswordSuccess,**

**RemoveLoginSuccess,**

**RemovePhoneSuccess,**

**Error**

**}**

**#endregion**

**}**

**}**

DEVELOPER’S GUIDE

Below are the functions that used in the Application

public DoctorsController() : This is the public constructor for the Doctors Controller. the Application database context was Initialized here.

public ActionResult Doctors() : it is a public function that returns a View with a list of all the doctors registered in this application.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor to avoid database openings.

Modules in QuestionsControllers:

public QuestionsController() : This is the public constructor for the Questions Controller. the Application database context was Initialized here.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor.

public ActionResult Index() : this is a public function that returns the view for the questions Page.

public ActionResult Add(Questions questions) : This is a public function that adds a new question submitted by a user in the application. it takes an Object parameter of Question Model Class then saves it in the Database.

public ActionResult AllQuestions() : it is a public function that returns a View with a list of all the questions asked by users in this application.

Modules in RepliesControllers:

public RepliesController() : This is the public constructor for the Replies Controller. the Application database context was Initialized here.

protected override void Dispose(bool disposing) : it is a protected function that was overridden from the controller class. it is used to close the Application database context opened in the constructor.

public ActionResult Index() : it is a public function that returns a View with a list of all the replies for a particular question asked by a user in this application.

public ActionResult Reply(int id) : It is a public function with a parameter Id of type Integer used to get the question then returns a view with the question gotten by the parameter passed and with a list of all the replies for the question.

public ActionResult ReplySave(Replies replies, int id): It is a public function with a parameter of type Object of the Replies Model Class and id of the question. it is used to save replies for a particular Question and returns a view with the question and a list of replies for the question.

public ActionResult ViewAll(int id): It is a public function with a parameter Id of type Integer used to get all the replies for a particular Question. It returns a view with the list of replies for the question.