Project Report

“QuestionsAreWelcome – An online Question Answer portal“

Summary:

QuestionsAreWelcome is a question-and-answer platform wherein questions are asked, answered and edited by the users. Here users can get answers of any questions in defined domain. Users can post their questions related to specific domains like Space, History, Literature, Places, World Affairs and Health and Fitness. Once approved, users who are expert in that domain can give the answers. A question once posted can be edited or deleted till it is not answered. An answer can be edited or deleted at any time. Every question/answer must go through an approver, who will approve or reject the question/answer. If approver finds that question is not of the domain specified, inappropriate or can hurt the sentiments of person, he/she has right to reject it with a justified reason. As soon as a question/answer is approved/rejected owner of the question/answer receives an email stating the status of it. Every user can like/dislike the question/answer. Users can give n number of answers to a question, but they can’t post same question or answer again. Admin can view dashboards and print the pdf of dashboards.

Roles supported:

User: Users need to register. Once registered, they can login to post a question. Users can view other users’ questions and answer them. They can edit/delete their questions and answers.

Approver: Approver can approve the question/answer or reject question/answer with rejection reason.

Admin: Admin is the owner/controller of the application. Admin can add approvers and view dashboards as well as download the pdf of dashboard.

Functionalities:

1) Admin:
   - Admin Login
   - Add Approver and its details
   - View Dashboards and download the pdf of dashboard. HashMaps are used to view dashboards and AbstractPdfView class is used to created PDF.
2) Approver:
   - Approver Login
   - View posts
   - Approve/Reject posts. An email is sent by the system once a question/answer is approved/rejected. JavaMail and SMTP is used to send email.
   - If rejected, add rejection reason specific to post
   - Update Profile
3) User:
   - User Registration
   - User Login
   - Post a question
   - Edit/Delete the question
   - Answer the questions
   - Edit/Delete the answers
   - Like/Dislike any question/answer
   - Update Profile

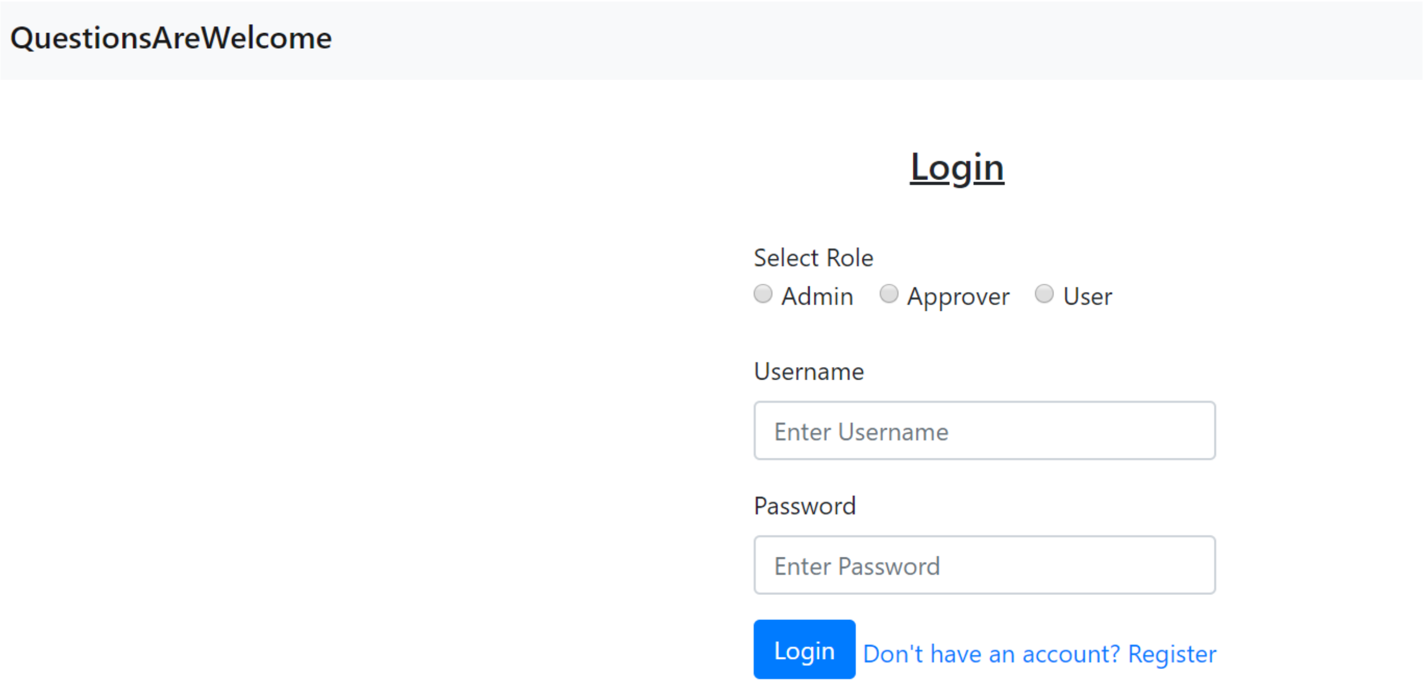
4) OneToOne, OneToMany and other associations are used to relate different Models.
5) Filter is used to prevent SQL injection.
6) Checks are made for duplicate users, questions and answers.
7) Front-end and back-end validations are performed for all user inputs.

Technology stack:

Technology used is Spring MVC and Hibernate.
Front End (Views): HTML5, CSS3, Bootstrap, JSP, Spring MVC Views, EL, JSTL and Custom Tags
Back End: Spring MVC Controllers, Hibernate, Spring Criteria, Spring Annotations
Database: Relational Database – MySQL

Screenshots:

Login Page



QuestionsAreWelcome

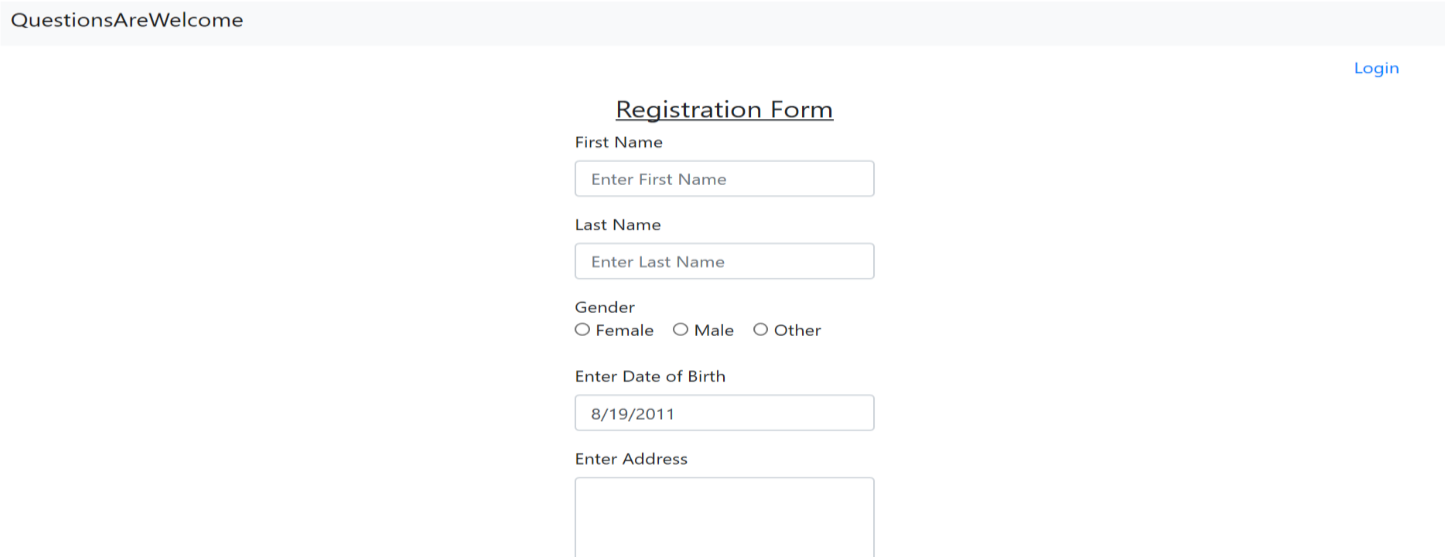
Login

Select Role
○ Admin ○ Approver ○ User

Username
Enter Username

Password
Enter Password

User Registration



QuestionsAreWelcome

Registration Form

First Name
Enter First Name

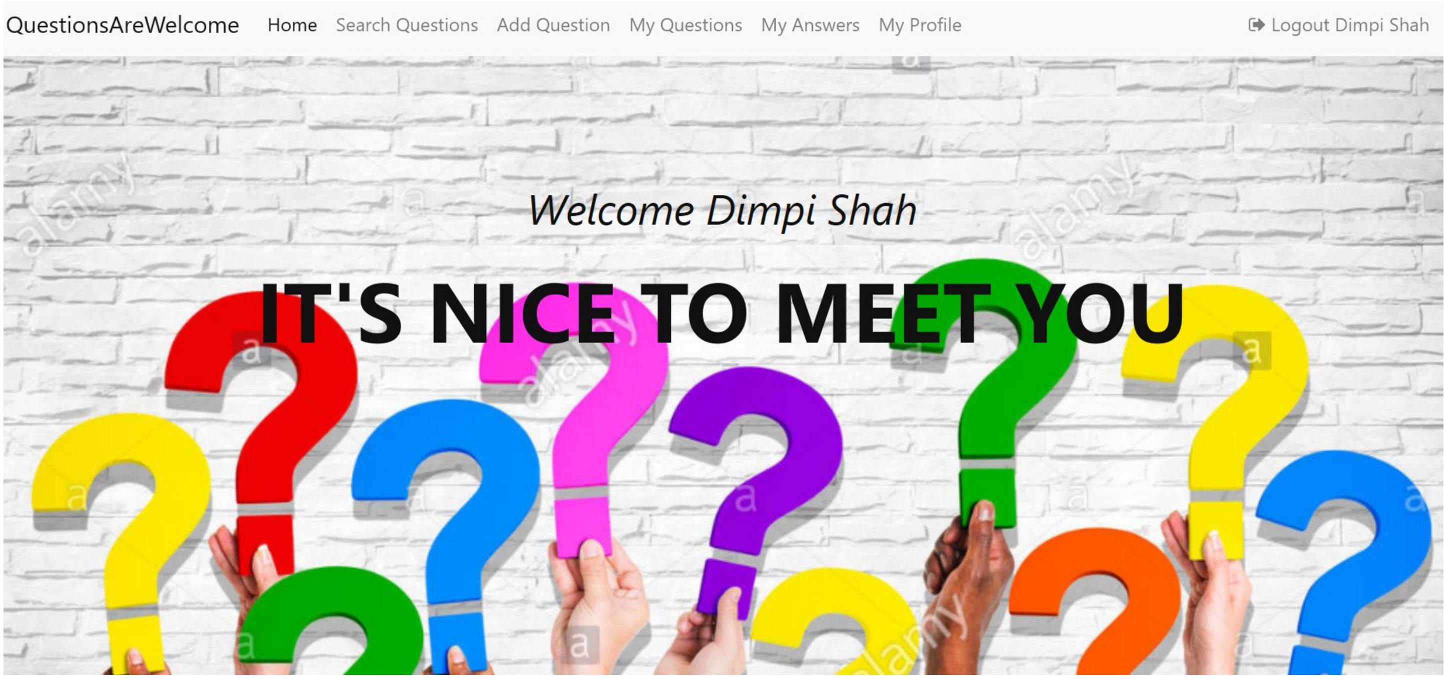
Last Name
Enter Last Name

Gender
○ Female ○ Male ○ Other

Enter Date of Birth
8/19/2011

Enter Address

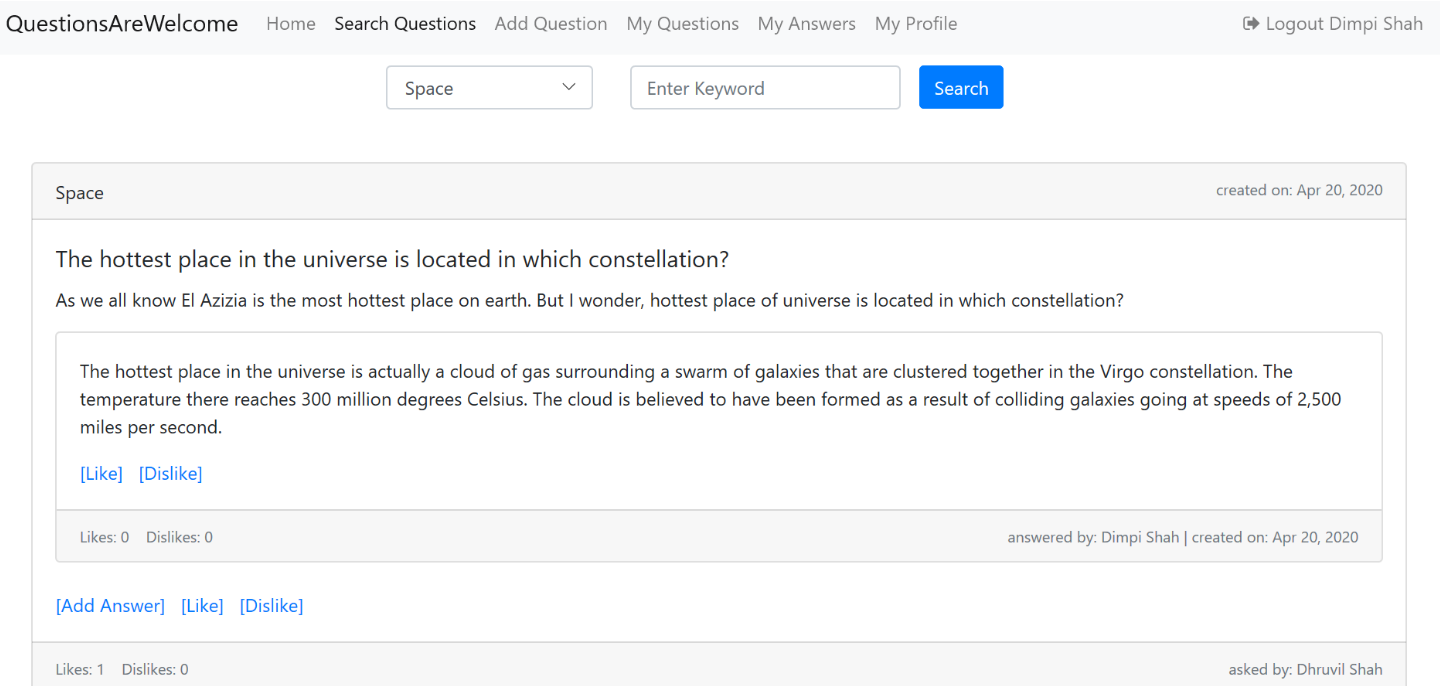
User Home Page



Welcome Dimpi Shah

IT'S NICE TO MEET YOU

User Search Questions Page



Space

created on: Apr 20, 2020

The hottest place in the universe is located in which constellation?

As we all know El Azizia is the most hottest place on earth. But I wonder, hottest place of universe is located in which constellation?

The hottest place in the universe is actually a cloud of gas surrounding a swarm of galaxies that are clustered together in the Virgo constellation. The temperature there reaches 300 million degrees Celsius. The cloud is believed to have been formed as a result of colliding galaxies going at speeds of 2,500 miles per second.

[Like] [Dislike]

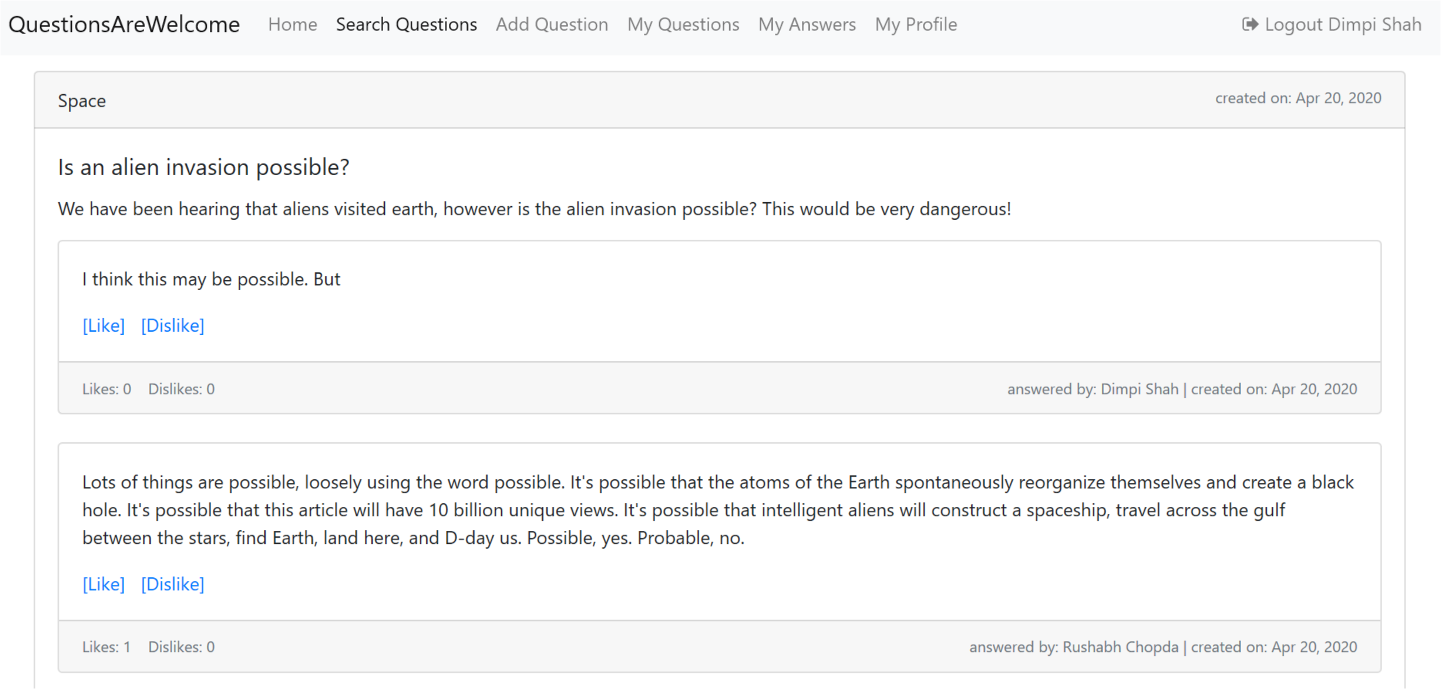
Likes: 0 Dislikes: 0

answered by: Dimpi Shah | created on: Apr 20, 2020

[Add Answer] [Like] [Dislike]

Likes: 1 Dislikes: 0

asked by: Dhruvil Shah



Space

created on: Apr 20, 2020

Is an alien invasion possible?

We have been hearing that aliens visited earth, however is the alien invasion possible? This would be very dangerous!

I think this may be possible. But

Likes: 0 Dislikes: 0

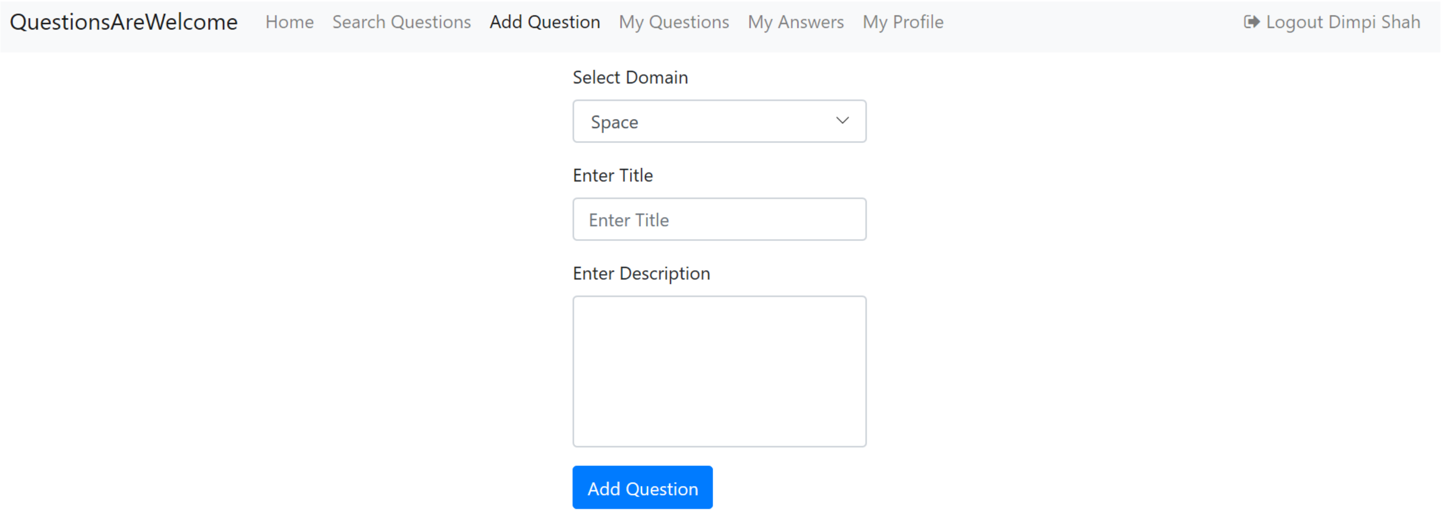
answered by: Dimpi Shah | created on: Apr 20, 2020

Lots of things are possible, loosely using the word possible. It's possible that the atoms of the Earth spontaneously reorganize themselves and create a black hole. It's possible that this article will have 10 billion unique views. It's possible that intelligent aliens will construct a spaceship, travel across the gulf between the stars, find Earth, land here, and D-day us. Possible, yes. Probable, no.

Likes: 1 Dislikes: 0

answered by: Rushabh Chopda | created on: Apr 20, 2020

User Add Question Page

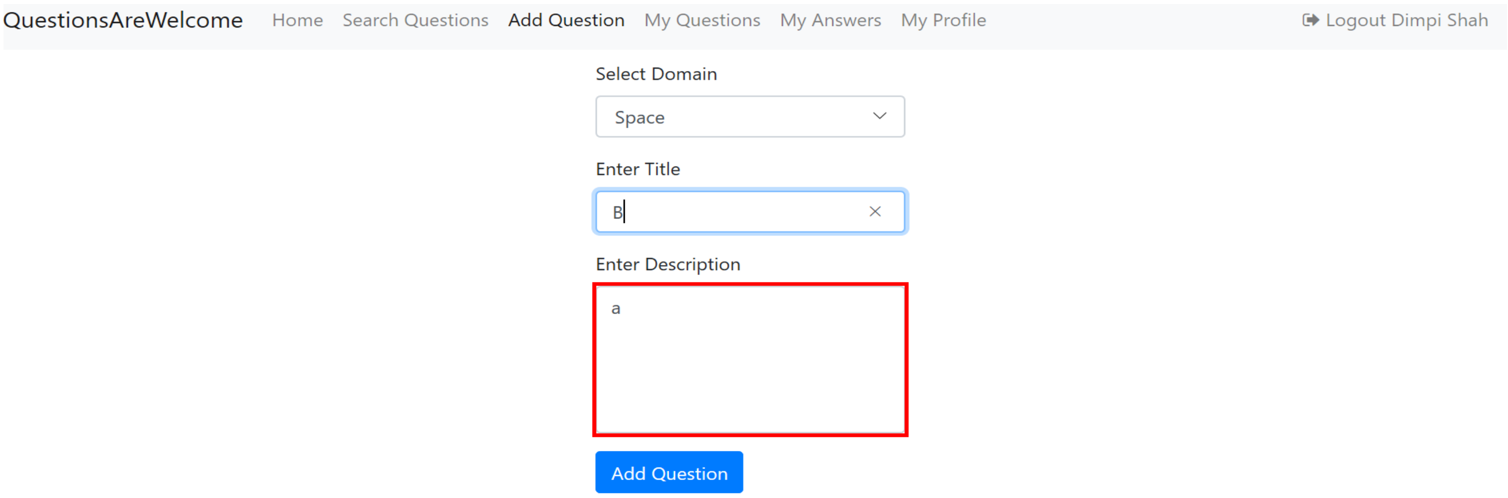


Select Domain: Space

Enter Title: Enter Title

Enter Description:

User Add Question Validations

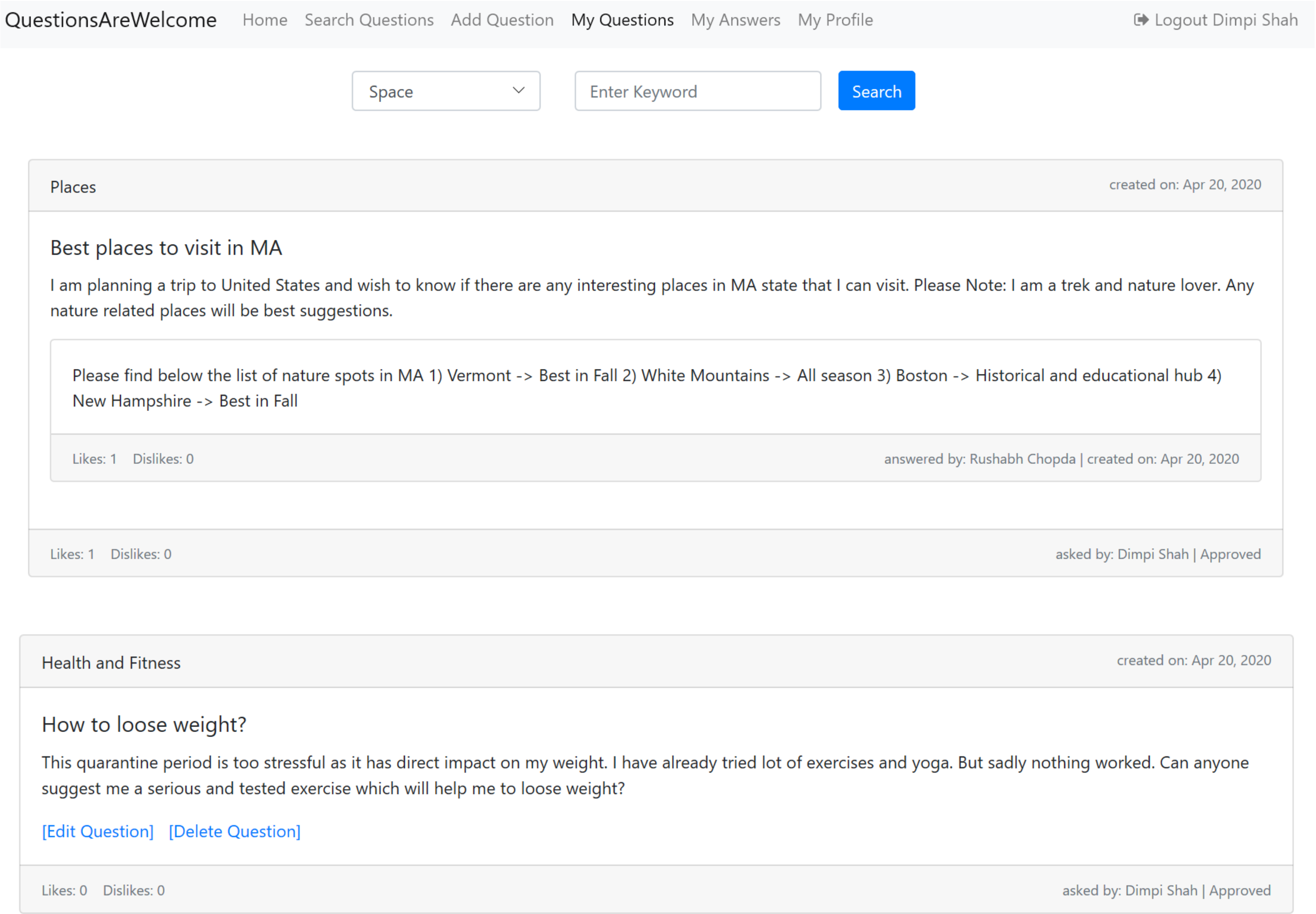


Select Domain: Space

Enter Title: B

Enter Description: a

User My Question List Page



Places

created on: Apr 20, 2020

Best places to visit in MA

I am planning a trip to United States and wish to know if there are any interesting places in MA state that I can visit. Please Note: I am a trek and nature lover. Any nature related places will be best suggestions.

Please find below the list of nature spots in MA 1) Vermont -> Best in Fall 2) White Mountains -> All season 3) Boston -> Historical and educational hub 4) New Hampshire -> Best in Fall

Likes: 1 Dislikes: 0

answered by: Rushabh Chopda | created on: Apr 20, 2020

Likes: 1 Dislikes: 0

asked by: Dimpi Shah | Approved

Health and Fitness

created on: Apr 20, 2020

How to loose weight?

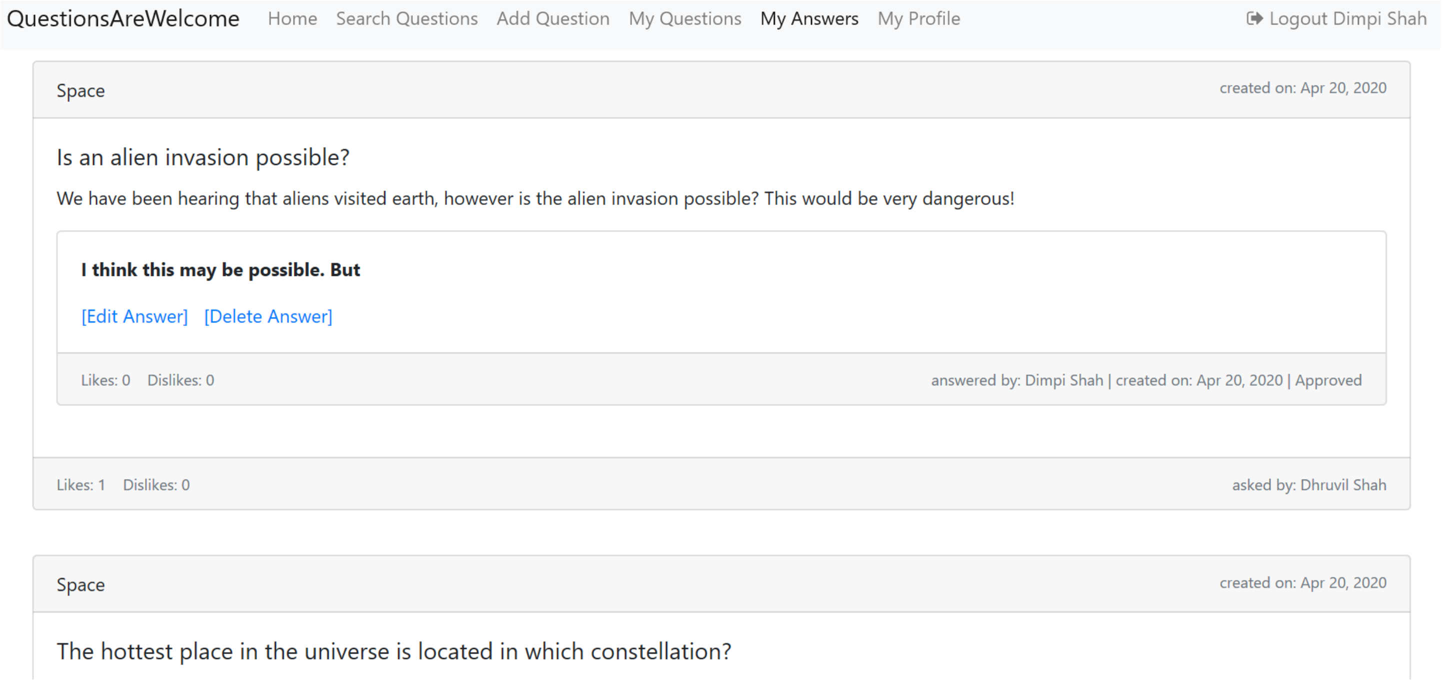
This quarantine period is too stressful as it has direct impact on my weight. I have already tried lot of exercises and yoga. But sadly nothing worked. Can anyone suggest me a serious and tested exercise which will help me to loose weight?

[Edit Question] [Delete Question]

Likes: 0 Dislikes: 0

asked by: Dimpi Shah | Approved

User My Answer List Page



Space

created on: Apr 20, 2020

Is an alien invasion possible?

We have been hearing that aliens visited earth, however is the alien invasion possible? This would be very dangerous!

I think this may be possible. But

[Edit Answer] [Delete Answer]

Likes: 0 Dislikes: 0

answered by: Dimpi Shah | created on: Apr 20, 2020 | Approved

Likes: 1 Dislikes: 0

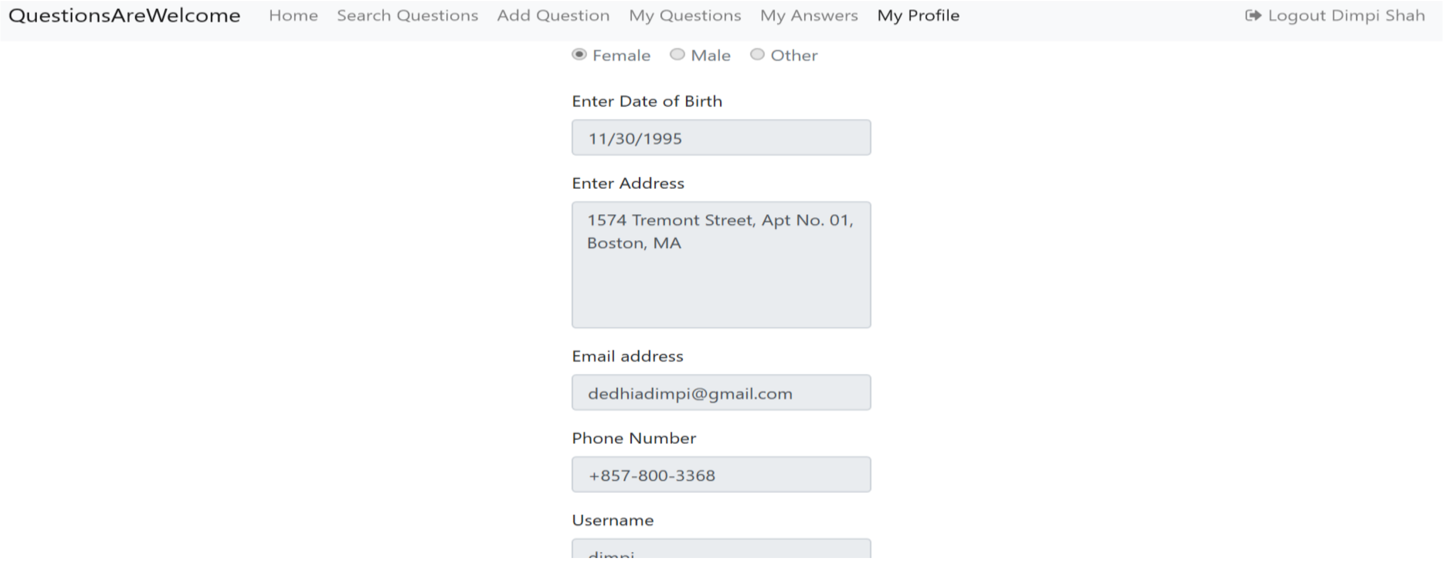
asked by: Dhruvil Shah

Space

created on: Apr 20, 2020

The hottest place in the universe is located in which constellation?

User My Profile Page



● Female ○ Male ○ Other

Enter Date of Birth

11/30/1995

Enter Address

1574 Tremont Street, Apt No. 01, Boston, MA

Email address

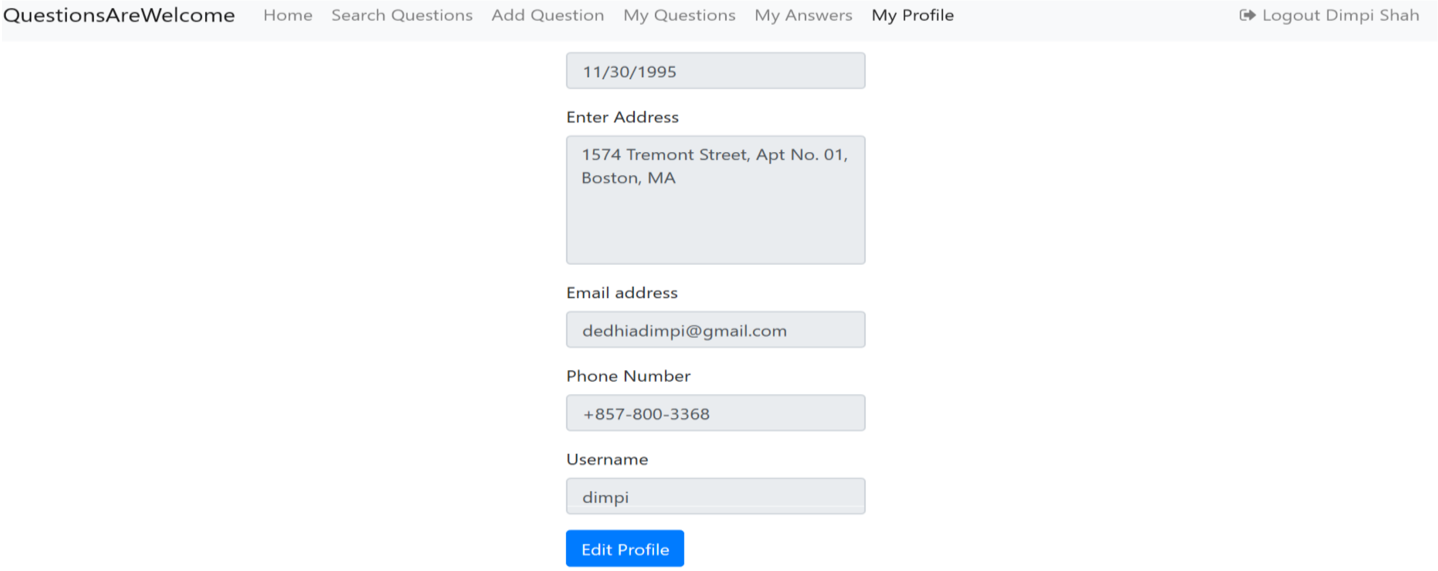
dedhiadimpi@gmail.com

Phone Number

+857-800-3368

Username

dimpi



11/30/1995

Enter Address

1574 Tremont Street, Apt No. 01, Boston, MA

Email address

dedhiadimpi@gmail.com

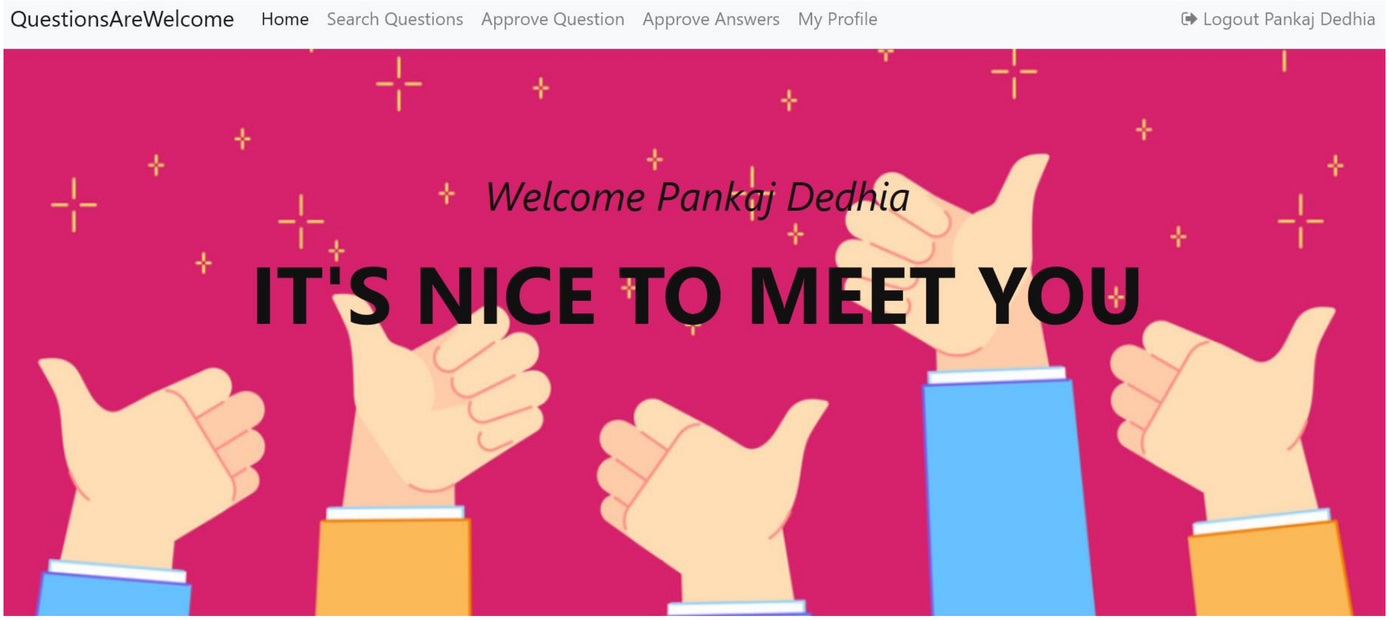
Phone Number

+857-800-3368

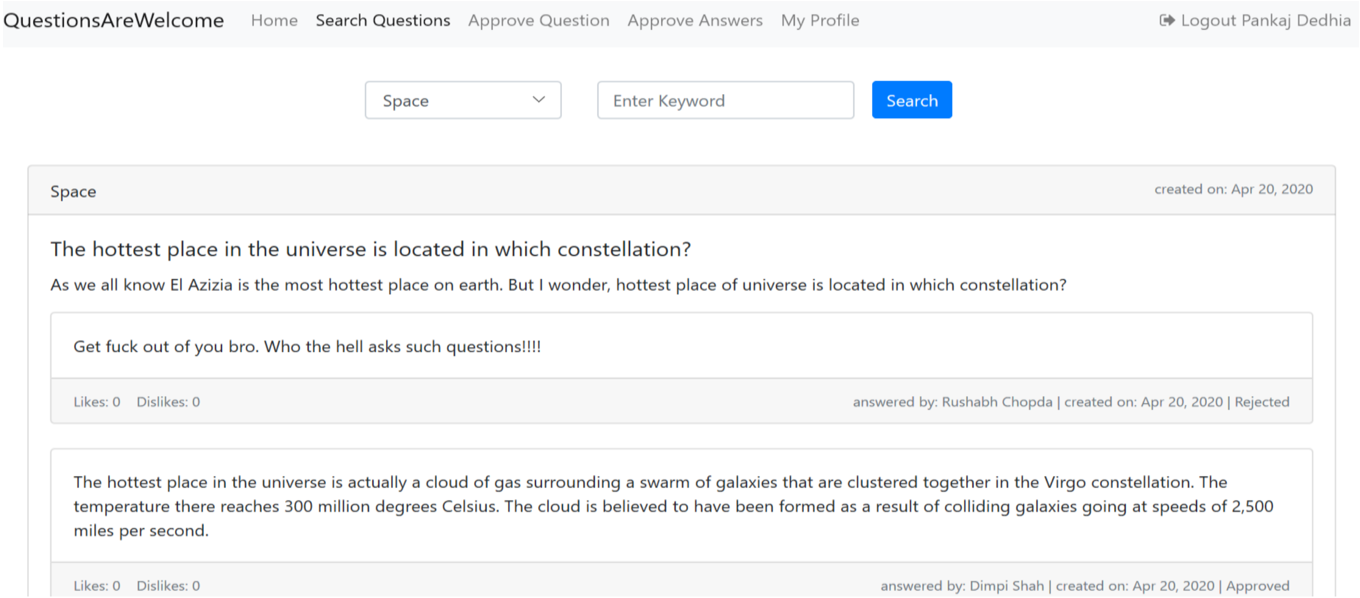
Username

dimpi

Approver Home Page



Approver Question & Answer List Page



Space

created on: Apr 20, 2020

The hottest place in the universe is located in which constellation?

As we all know El Azizia is the most hottest place on earth. But I wonder, hottest place of universe is located in which constellation?

Get fuck out of you bro. Who the hell asks such questions!!!!

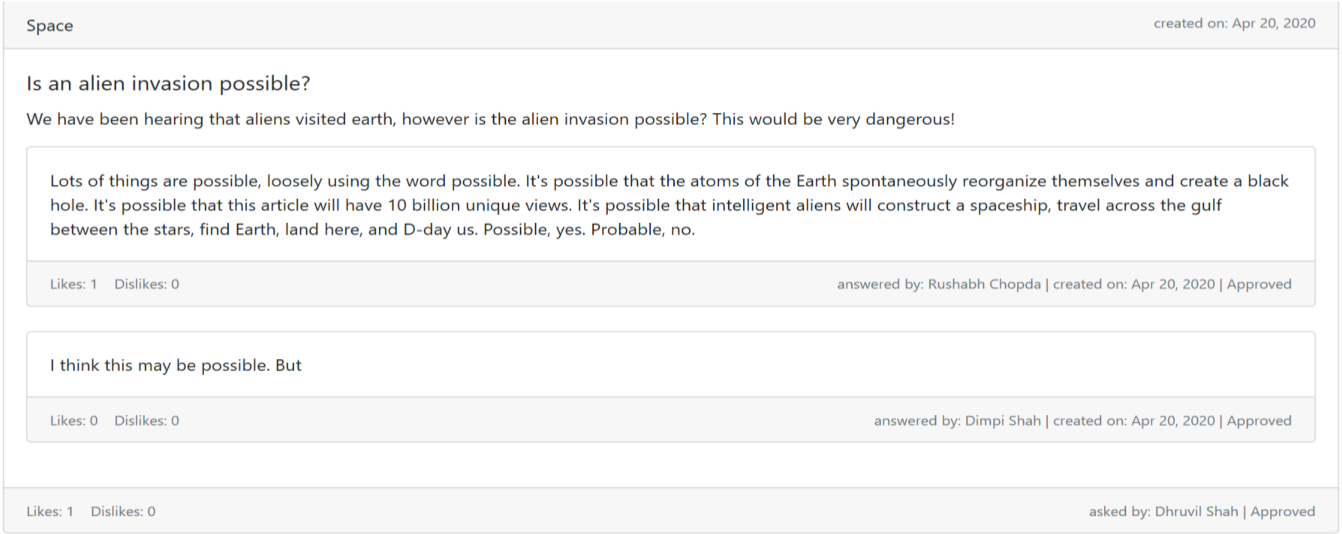
Likes: 0 Dislikes: 0

answered by: Rushabh Chopda | created on: Apr 20, 2020 | Rejected

The hottest place in the universe is actually a cloud of gas surrounding a swarm of galaxies that are clustered together in the Virgo constellation. The temperature there reaches 300 million degrees Celsius. The cloud is believed to have been formed as a result of colliding galaxies going at speeds of 2,500 miles per second.

Likes: 0 Dislikes: 0

answered by: Dimpi Shah | created on: Apr 20, 2020 | Approved



Space

created on: Apr 20, 2020

Is an alien invasion possible?

We have been hearing that aliens visited earth, however is the alien invasion possible? This would be very dangerous!

Lots of things are possible, loosely using the word possible. It's possible that the atoms of the Earth spontaneously reorganize themselves and create a black hole. It's possible that this article will have 10 billion unique views. It's possible that intelligent aliens will construct a spaceship, travel across the gulf between the stars, find Earth, land here, and D-day us. Possible, yes. Probable, no.

Likes: 1 Dislikes: 0

answered by: Rushabh Chopda | created on: Apr 20, 2020 | Approved

I think this may be possible. But

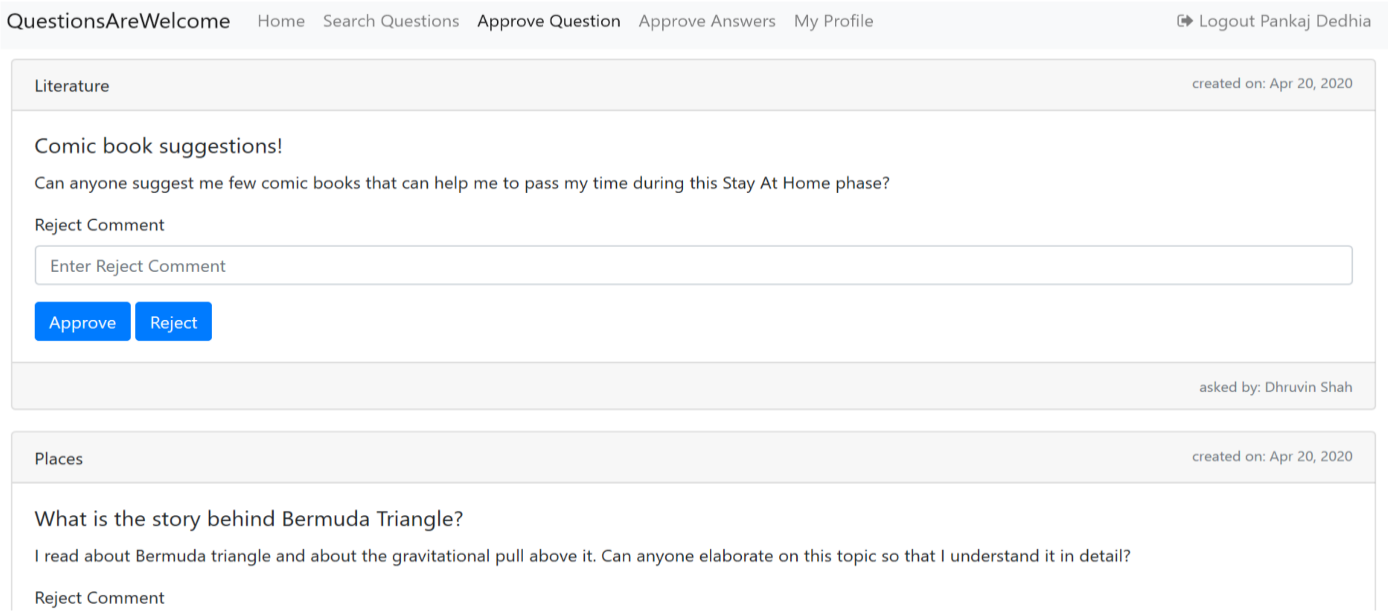
Likes: 0 Dislikes: 0

answered by: Dimpi Shah | created on: Apr 20, 2020 | Approved

Likes: 1 Dislikes: 0

asked by: Dhruvil Shah | Approved

Approver Approve Question Page



Literature

created on: Apr 20, 2020

Comic book suggestions!

Can anyone suggest me few comic books that can help me to pass my time during this Stay At Home phase?

Reject Comment

Enter Reject Comment

asked by: Dhruvin Shah

Places

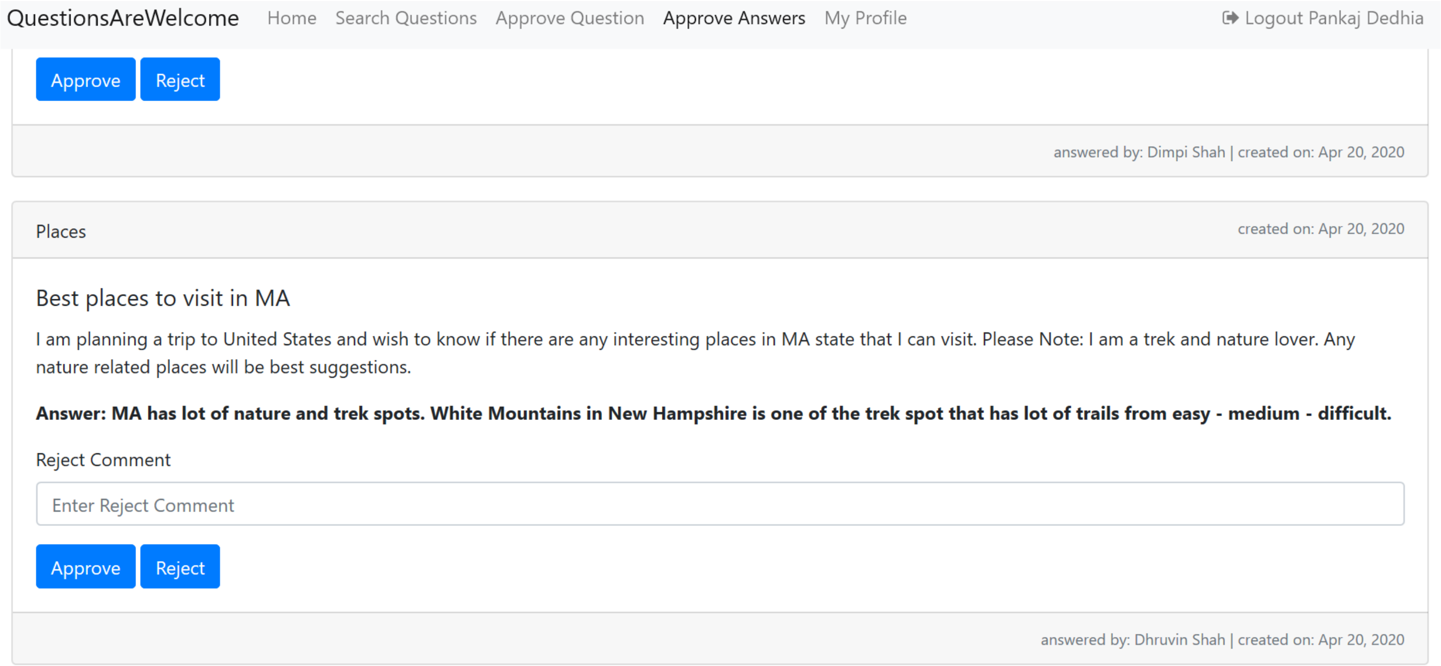
created on: Apr 20, 2020

What is the story behind Bermuda Triangle?

I read about Bermuda triangle and about the gravitational pull above it. Can anyone elaborate on this topic so that I understand it in detail?

Reject Comment

Approver Approve Answer Page



answered by: Dimpi Shah | created on: Apr 20, 2020

Places

created on: Apr 20, 2020

Best places to visit in MA

I am planning a trip to United States and wish to know if there are any interesting places in MA state that I can visit. Please Note: I am a trek and nature lover. Any nature related places will be best suggestions.

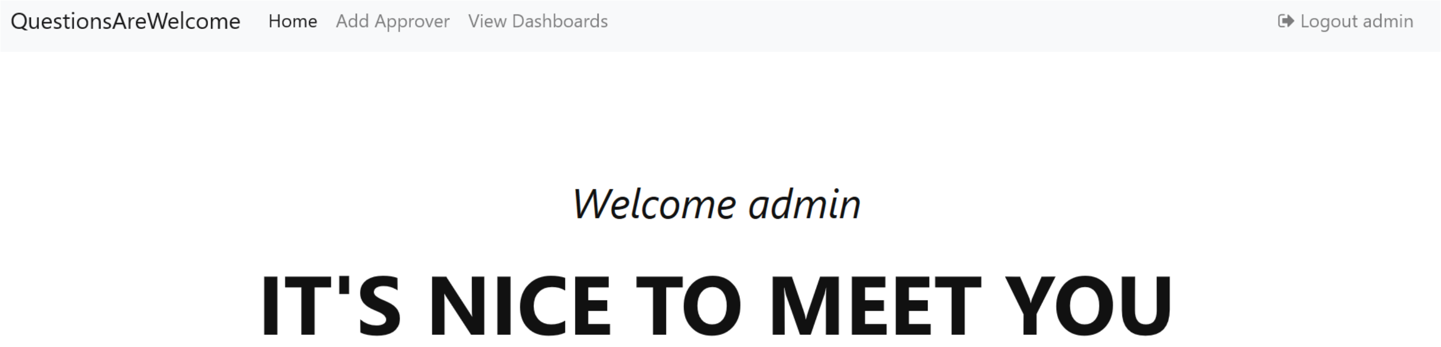
Answer: MA has lot of nature and trek spots. White Mountains in New Hampshire is one of the trek spot that has lot of trails from easy - medium - difficult.

Reject Comment

Enter Reject Comment

answered by: Dhruvin Shah | created on: Apr 20, 2020

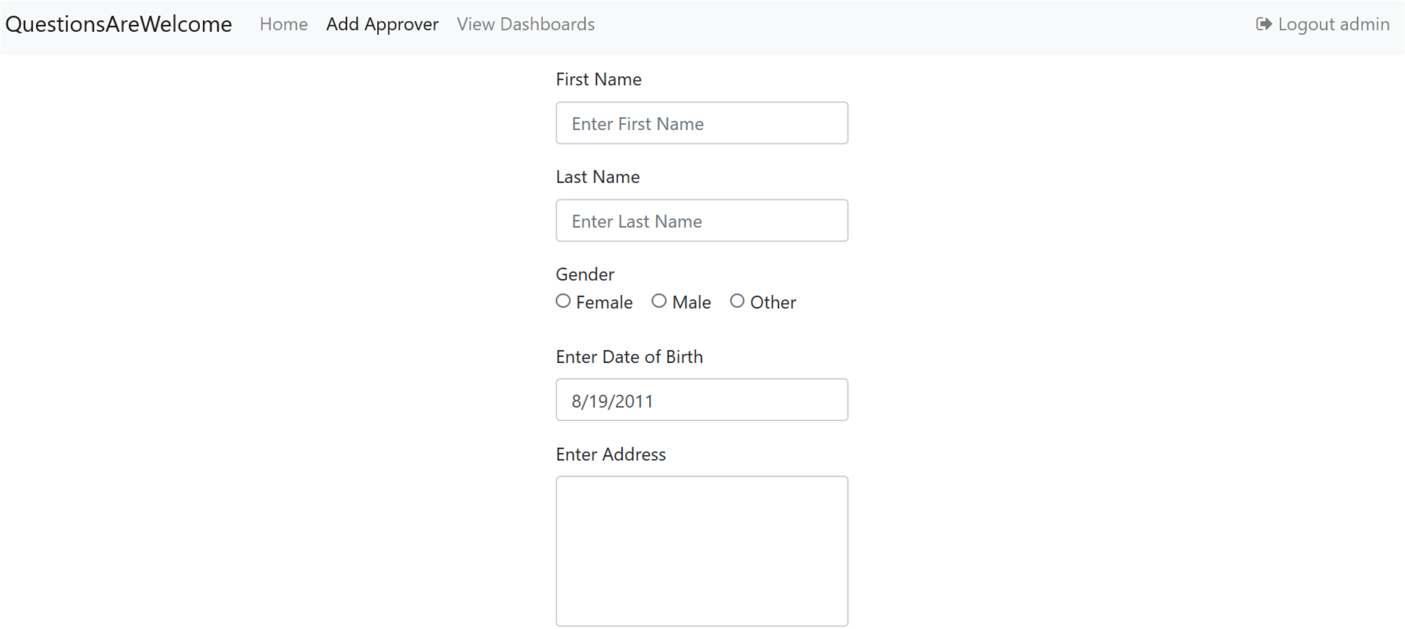
Admin Home Page



Welcome admin

IT'S NICE TO MEET YOU

Admin Add Approver Page



First Name

Enter First Name

Last Name

Enter Last Name

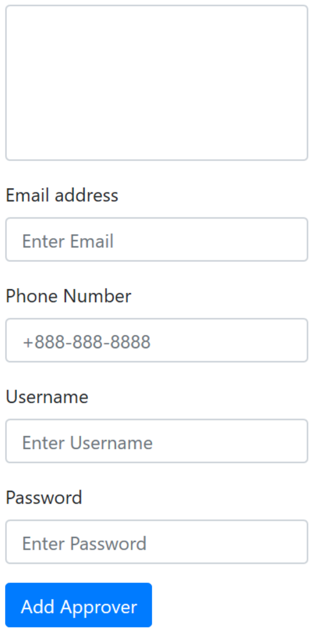
Gender

○ Female ○ Male ○ Other

Enter Date of Birth

8/19/2011

Enter Address



Email address

Enter Email

Phone Number

+888-888-8888

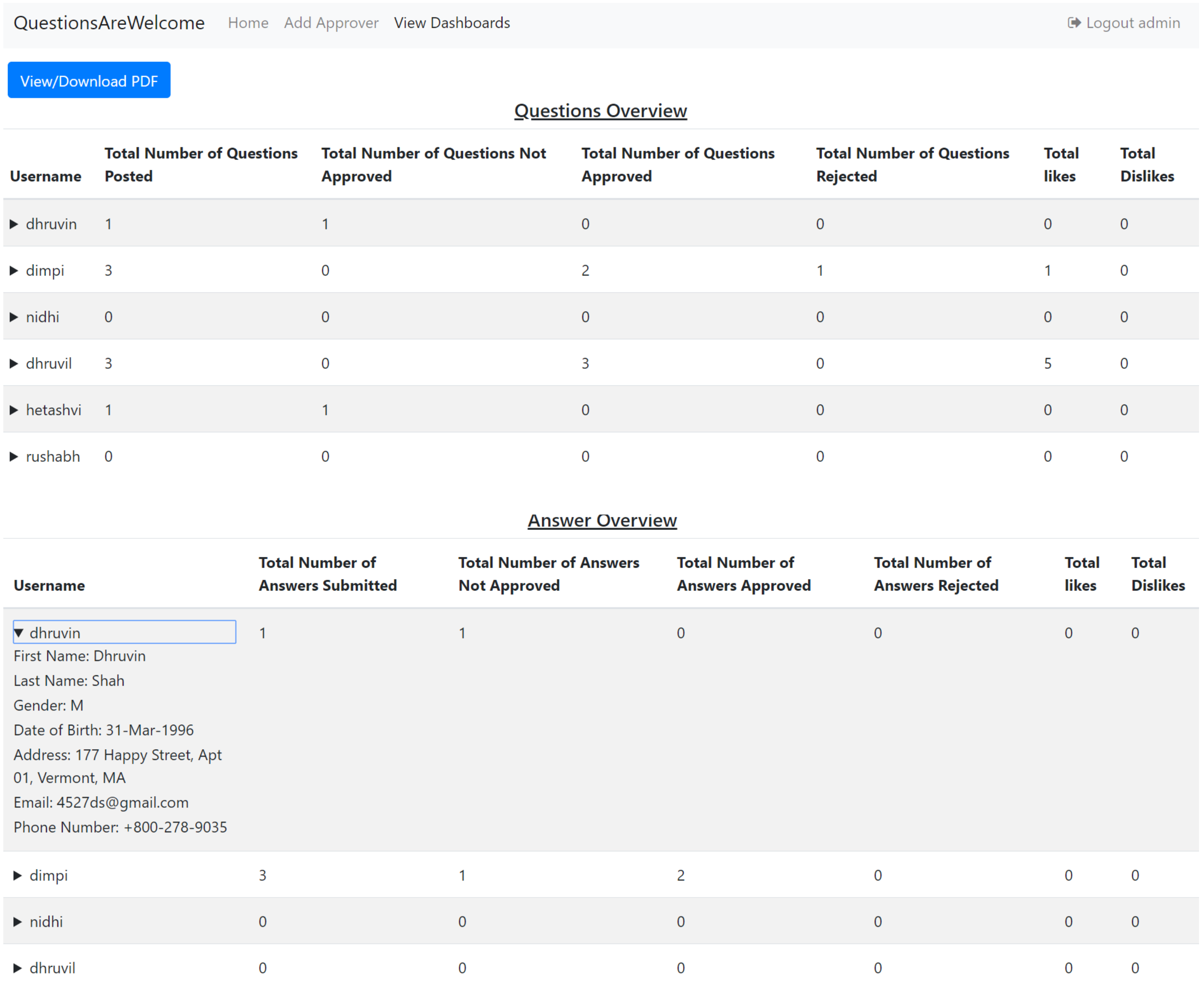
Username

Enter Username

Password

Enter Password

Admin View Dashboard Page

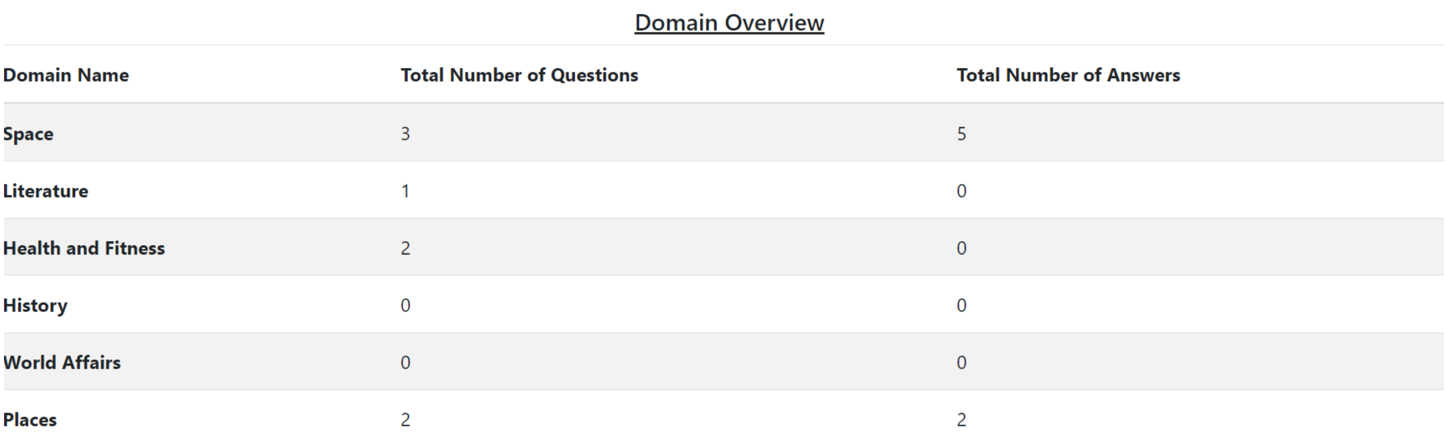


Questions Overview

Username	Total Number of Questions Posted	Total Number of Questions Not Approved	Total Number of Questions Approved	Total Number of Questions Rejected	Total likes	Total Dislikes
▶ dhruvin	1	1	0	0	0	0
▶ dimpi	3	0	2	1	1	0
▶ nidhi	0	0	0	0	0	0
▶ dhruvil	3	0	3	0	5	0
▶ hetashvi	1	1	0	0	0	0
▶ rushabh	0	0	0	0	0	0

Answer Overview

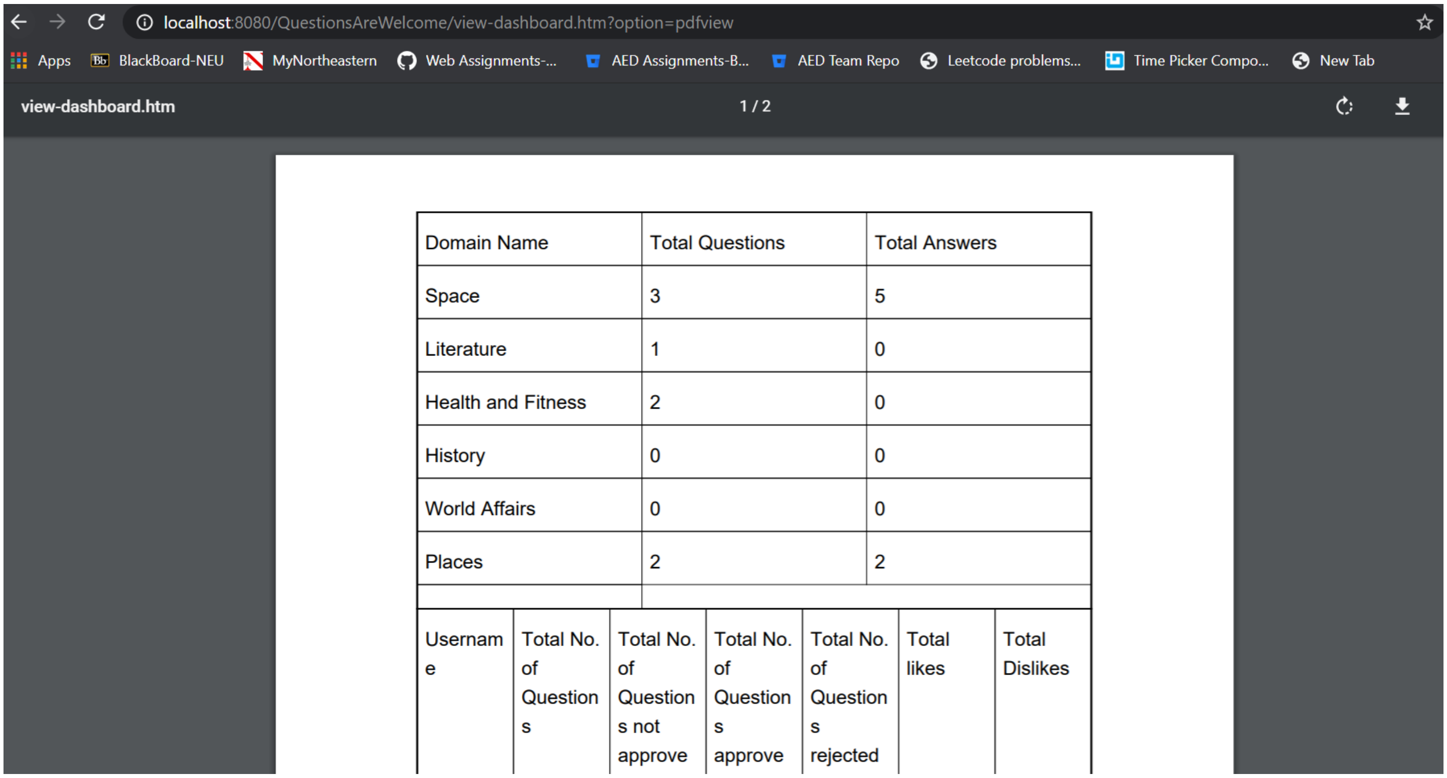
Username	Total Number of Answers Submitted	Total Number of Answers Not Approved	Total Number of Answers Approved	Total Number of Answers Rejected	Total likes	Total Dislikes
▼ dhruvin First Name: Dhruvin Last Name: Shah Gender: M Date of Birth: 31-Mar-1996 Address: 177 Happy Street, Apt 01, Vermont, MA Email: 4527ds@gmail.com Phone Number: +800-278-9035	1	1	0	0	0	0
▶ dimpi	3	1	2	0	0	0
▶ nidhi	0	0	0	0	0	0
▶ dhruvil	0	0	0	0	0	0



Domain Overview

Domain Name	Total Number of Questions	Total Number of Answers
Space	3	5
Literature	1	0
Health and Fitness	2	0
History	0	0
World Affairs	0	0
Places	2	2

Admin PDF View of Dashboard



Domain Name	Total Questions	Total Answers
Space	3	5
Literature	1	0
Health and Fitness	2	0
History	0	0
World Affairs	0	0
Places	2	2

Username	Total No. of Questions	Total No. of Questions not approve	Total No. of Questions approve	Total No. of Questions rejected	Total likes	Total Dislikes

Controller Code:

1) Login Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controllers.LoginController;

import Helper.Validation;
import Models.Admin.Admin;
import Models.Approver.Approver;
import LoginDao.LoginDao;
import Models.User.User;
import java.io.PrintWriter;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;

/**
 *
 * @author dedhi
 */
public class LoginController extends AbstractController{

    @Override
    protected ModelAndView handleRequestInternal(HttpServletRequest request, HttpServletResponse response) throws
Exception {
        ModelAndView mv = null;
        Validation validation = new Validation();
        String option = request.getParameter("option") == null ? "" : request.getParameter("option");
        if (option == null || option == "") {
            mv = new ModelAndView("login");
            return mv;
        } else if(option.equals("login")){
            String role = request.getParameter("role") == null? "none": request.getParameter("role");
            String username = request.getParameter("username");
            String password = request.getParameter("password");
            LoginDao loginDao = new LoginDao();
            if(role.equals("admin")){
                ArrayList<Admin> admin = loginDao.getAdminLoginDetails(username, password);
                if(admin.size() > 0){
                    request.getSession().setAttribute("user", admin.get(0));
                    request.getSession().setAttribute("role", "admin");
                    mv = new ModelAndView("admin-home", "role", admin.get(0));
                    return mv;
                }else{
                    mv = new ModelAndView("login");
                    mv.addObject("error","User or Password is Invalid. Kindly Register if you don't have an account!");
                    return mv;
                }
            }else if(role.equals("approver")){
                ArrayList<Approver> approver = loginDao.getApproverLoginDetails(username, password);
                if(approver.size() > 0){
                    request.getSession().setAttribute("user", approver.get(0));
                    request.getSession().setAttribute("role", "approver");
                    mv = new ModelAndView("approver-home", "role", approver.get(0));
                    return mv;
                }else{
                    mv = new ModelAndView("login");
                    mv.addObject("error","User or Password is Invalid. Kindly Register if you don't have an account!");
```

```
          return mv;
        }
      }else if(role.equals("user")){
        ArrayList<User> user = loginDao.getUserLoginDetails(username, password);
        if(user.size() > 0){
          request.getSession().setAttribute("user", user.get(0));
          request.getSession().setAttribute("role", "user");
          mv = new ModelAndView("user-home", "role", user.get(0));
          return mv;
        }else{
          mv = new ModelAndView("login");
          mv.addObject("error","User or Password is Invalid. Kindly Register if you don't have an account!");
          return mv;
        }

      }else{
        mv = new ModelAndView("login");
        mv.addObject("error","Please specify a role!");
        return mv;
      }
    }else if(option.equals("register")){
      LoginDao loginDao = new LoginDao();
      String firstname = request.getParameter("firstname");
      String lastname = request.getParameter("lastname");
      String gender = request.getParameter("gender") == null ? "" : request.getParameter("gender");
      Date date = null;
      String address = request.getParameter("address");
      String email = request.getParameter("email");
      String phone = request.getParameter("phone");
      String username = request.getParameter("username");
      String password = request.getParameter("password");
      try{
        date = new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("dateofbirth").substring(0, 9));
      }catch(Exception e){
        mv = new ModelAndView("register-user");
        mv.addObject("error","Please enter correct date!");
        mv.addObject("firstname",firstname);
        mv.addObject("lastname",lastname);
        mv.addObject("date", new Date());
        mv.addObject("address",address);
        mv.addObject("email",email);
        mv.addObject("phone",phone);
        mv.addObject("username",username);
        mv.addObject("password",password);
        return mv;
      }
      if(date.compareTo(new Date()) > 0){
        mv = new ModelAndView("register-user");
        mv.addObject("error","Please enter a date less than today's date!");
        mv.addObject("firstname",firstname);
        mv.addObject("lastname",lastname);
        mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
        mv.addObject("address",address);
```

```
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }
        if(gender.equals("")){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter your Gender!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(!validation.onlyLettersValidation(firstname) || firstname.trim().length() < 0 || firstname.trim().length() >
50){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter valid First Name! First name cannot have more than 50 letters");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(!validation.onlyLettersValidation(lastname) || lastname.trim().length() < 0 || lastname.trim().length() >
50){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter valid Last Name! Last name cannot have more than 50 letters");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(address.trim().length() < 8 || address.trim().length() > 100){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter valid Address! Address cannot be less than 8 letters or more than 100
letters!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(!validation.phoneNumberValidation(phone) || phone.trim().length() > 20){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter valid Phone Number!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(username.trim().length() < 5 || username.trim().length() > 10){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Username should be minimum of 5 characters and maximum of 10 characters!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(validation.usernameWithSpacesValidation(username)){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Please enter valid Username!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(!validation.passwordPatternCorrect(password) || password.trim().length() < 6 || password.trim().length()
> 10){
          mv = new ModelAndView("register-user");
          mv.addObject("error","Password should have 1 small-case letter, 1 Capital letter, 1 digit, 1 special character
and the length should be between 6-10 characters");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }
        int r = loginDao.checkUser(email, username);
        if(r == 1){
          mv = new ModelAndView("register-user");
          mv.addObject("error","User already exist!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }
        int result = loginDao.registerUser(firstname, lastname, gender, date, address, email, phone, username,
password);

        if(result == 1){
          mv = new ModelAndView("login","error","User registered successfully!");
          return mv;
        }else {
          mv = new ModelAndView("register-user");
          mv.addObject("error","Something went wrong. Please try again");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }

      }else if(option.equals("logout")){
            if(request.getSession().getAttribute("user") == null){
        mv = new ModelAndView("login");
        return mv;
      }
        request.getSession().invalidate();
        mv = new ModelAndView("login");
        return mv;
      }else if(option.equals("addApprover")){
            if(request.getSession().getAttribute("user") == null){
        mv = new ModelAndView("login");
        return mv;
      }
        LoginDao loginDao = new LoginDao();
        String firstname = request.getParameter("firstname");
        String lastname = request.getParameter("lastname");
        String gender = request.getParameter("gender");
```

```
        Date date = new Date();
//         Date date = new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("dateofbirth").substring(0, 9));
        String address = request.getParameter("address");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        try{
           date = new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("dateofbirth").substring(0, 9));
        }catch(Exception e){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter correct date!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new Date());
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }
        if(date.compareTo(new Date()) > 0){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter a date less than today's date!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }
        if(gender.equals("")){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter your Gender!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }else if(!validation.onlyLettersValidation(firstname) || firstname.trim().length() < 0 || firstname.trim().length() >
50){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter valid First Name! First name cannot have more than 50 letters");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }else if(!validation.onlyLettersValidation(lastname) || lastname.trim().length() < 0 || lastname.trim().length() >
50){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter valid Last Name! Last name cannot have more than 50 letters");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }else if(address.trim().length() < 8 || address.trim().length() > 100){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter valid Address! Address cannot be less than 8 letters or more than 100
letters!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }else if(!validation.phoneNumberValidation(phone) || phone.trim().length() > 20){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Please enter valid Phone Number!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
           mv.addObject("email",email);
           mv.addObject("phone",phone);
           mv.addObject("username",username);
           mv.addObject("password",password);
           return mv;
        }else if(username.trim().length() < 5 || username.trim().length() > 10){
           mv = new ModelAndView("add-approver");
           mv.addObject("error","Username should be minimum of 5 characters and maximum of 10 characters!");
           mv.addObject("firstname",firstname);
           mv.addObject("lastname",lastname);
           mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
           mv.addObject("address",address);
```

```
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(validation.usernameWithSpacesValidation(username)){
          mv = new ModelAndView("add-approver");
          mv.addObject("error","Please enter valid Username!");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }else if(!validation.passwordPatternCorrect(password) || password.trim().length() < 6 || password.trim().length()
> 10){
          mv = new ModelAndView("add-approver");
          mv.addObject("error","Password should have 1 small-case letter, 1 Capital letter, 1 digit, 1 special character
and the length should be between 6-10 characters");
          mv.addObject("firstname",firstname);
          mv.addObject("lastname",lastname);
          mv.addObject("date", new SimpleDateFormat("yyyy-MM-dd").format(date));
          mv.addObject("address",address);
          mv.addObject("email",email);
          mv.addObject("phone",phone);
          mv.addObject("username",username);
          mv.addObject("password",password);
          return mv;
        }

        int result = loginDao.addApprover(firstname, lastname, gender, date, address, email, phone, username,
password);
        if(result == 1){
          mv = new ModelAndView("add-approver");
          mv.addObject("firstname","");
          mv.addObject("lastname","");
          mv.addObject("date", new Date());
          mv.addObject("address","");
          mv.addObject("email","");
          mv.addObject("phone","");
          mv.addObject("username","");
          mv.addObject("password","");
          mv.addObject("error","Approved registered successfully!");
          return mv;
        }else {
          mv = new ModelAndView("add-approver");
          mv.addObject("firstname","");
          mv.addObject("lastname","");
          mv.addObject("date", new Date());
          mv.addObject("address","");
```

```
                mv.addObject("email","");
                mv.addObject("phone","");
                mv.addObject("username","");
                mv.addObject("password","");
                mv.addObject("error","Something went wrong. Please try again later!");
                return mv;
            }

        }
        return mv;
    }

}
```

2) Admin Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controllers.AdminController;

import Dao.DashboardDao.DashboardDao;
import Models.User.User;
import java.util.ArrayList;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;

/**
 *
 * @author dedhi
 */
public class AdminController extends AbstractController{

    @Override
    protected ModelAndView handleRequestInternal(HttpServletRequest request, HttpServletResponse response) throws
Exception {
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        DashboardDao dashboardDao = new DashboardDao();
            Map<String,ArrayList<Object>> questionMap = dashboardDao.getUserDetailsWithQuestions();
            Map<String,ArrayList<Object>> answerMap = dashboardDao.getUserDetailsWithAnswers();
            Map<String, ArrayList<Integer>> domainMap = dashboardDao.getDomainDetails();
        if(request.getParameter("option").equals("view")){
            mv = new ModelAndView("view-dashboard");
            mv.addObject("questionMap",questionMap);
            mv.addObject("answerMap",answerMap);
            mv.addObject("domainMap",domainMap);

        return mv;
        }if(request.getParameter("option").equals("pdfview")){
            mv = new ModelAndView("pdfview-dashboard");
            mv.addObject("questionMap",questionMap);
            mv.addObject("answerMap",answerMap);
            mv.addObject("domainMap",domainMap);

        return mv;
        }

        return mv;
    }

}
```

3) Approver Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controllers.ApproverController;

import Dao.ApproverDao.ApproverDao;
import Helper.ApproveEmail;
import Models.Answer.Answer;
import Models.Approver.Approver;
import Models.Question.Question;
import Models.User.User;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;

/**
 *
 * @author dedhi
 */
public class ApproverController extends AbstractController{

    @Override
    protected ModelAndView handleRequestInternal(HttpServletRequest req, HttpServletResponse resp) throws
Exception {
        ModelAndView mv = null;
        if(req.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        ApproverDao approverDao = new ApproverDao();
        Approver approver = (Approver) req.getSession().getAttribute("user");

        if(req.getParameter("option").equals("question")){
            if(req.getParameter("action").equals("Approve")){
            long question_id = Long.parseLong(req.getParameter("question_id"));
            Question result = approverDao.approveQuestions(question_id, approver,"");
            ArrayList<Question> questionList = approverDao.getQuestionListForApproval();
            mv = new ModelAndView("approve-questions");
            mv.addObject("questionList",questionList);

            if(result != null){
              String configEmail = "email-bean.xml";
                    ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(configEmail);
                    // @Service call
                    ApproveEmail approveEmail = (ApproveEmail) context.getBean("approveEmail");
                    String toAddr = result.getUser().getEmail();
                    String fromAddr = "dedhiadimpi@gmail.com";
                    // email subject
                    String subject = "Good news from QuestonsAreWelcome!";
                    // email body
                    String body = "Hi "+result.getUser().getFirstname()+" "+result.getUser().getLastname()+",\n\nYour
Question '"+result.getQuestion_title()+"' is Approved and Posted.";
                    approveEmail.sendEmail(toAddr, fromAddr, subject, body);
              mv.addObject("error","Question Approved successfully!");
              mv.addObject("reject","");
              return mv;
            }else{
              mv.addObject("reject","");
              mv.addObject("error","Something went wrong. Please try again later!");
              return mv;
            }

        }else if(req.getParameter("action").equals("Reject")){
            String reject_comment = req.getParameter("reject_comment");
            if(reject_comment == null || reject_comment == "" || reject_comment.trim().length() < 5 ||
reject_comment.trim().length() > 4000){
              ArrayList<Question> questionList = approverDao.getQuestionListForApproval();
              mv = new ModelAndView("approve-questions");
              mv.addObject("questionList",questionList);
              mv.addObject("reject",reject_comment);
              mv.addObject("error","Please enter a valid rejection reason! Reject reason cannot be less than 5 or more than
4000 characters");
              return mv;
            }else{
              long question_id = Long.parseLong(req.getParameter("question_id"));
              Question result = approverDao.approveQuestions(question_id, approver,reject_comment);
              ArrayList<Question> questionList = approverDao.getQuestionListForApproval();
              mv = new ModelAndView("approve-questions");
              mv.addObject("questionList",questionList);

              if(result != null){
                String configEmail = "email-bean.xml";
                ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(configEmail);
                // @Service call
                ApproveEmail approveEmail = (ApproveEmail) context.getBean("approveEmail");
                String toAddr = result.getUser().getEmail();
                String fromAddr = "dedhiadimpi@gmail.com";
                // email subject
                String subject = "Bad news from QuestonsAreWelcome!";
                // email body
                String body = "Hi "+result.getUser().getFirstname()+" "+result.getUser().getLastname()+",\n\nYour Question
'"+result.getQuestion_title()+"' is Rejected with following reason\n"+result.getReject_comment();
                approveEmail.sendEmail(toAddr, fromAddr, subject, body);
                mv.addObject("reject","");
                mv.addObject("error","Question Rejected successfully!");
                return mv;
              }else{
                mv.addObject("reject",reject_comment);
                mv.addObject("error","Something went wrong. Please try again later!");
                return mv;
              }
            }
        }
        }else if(req.getParameter("option").equals("answer")){
            if(req.getParameter("action").equals("Approve")){
            long answer_id = Long.parseLong(req.getParameter("answer_id"));
            Answer result = approverDao.approveAnswers(answer_id, approver,"");
            ArrayList<Answer> answerList = approverDao.getAnswerListForApproval();
            mv = new ModelAndView("approve-answers");
            mv.addObject("answerList",answerList);
            mv.addObject("reject","");
            if(result != null){
              String configEmail = "email-bean.xml";
                    ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(configEmail);
                    // @Service call
                    ApproveEmail approveEmail = (ApproveEmail) context.getBean("approveEmail");
                    String toAddr = result.getUser().getEmail();
                    String fromAddr = "dedhiadimpi@gmail.com";
                    // email subject
                    String subject = "Good news from QuestonsAreWelcome!";
                    // email body
                    String body = "Hi "+result.getUser().getFirstname()+" "+result.getUser().getLastname()+",\n\nYour
Answer '"+result.getAnswer_description()+"' for Question '"+result.getQuestion().getQuestion_title()+"' is Approved and
Posted";
                    approveEmail.sendEmail(toAddr, fromAddr, subject, body);
              mv.addObject("error","Answer Approved successfully!");
              return mv;
            }else{
              mv.addObject("error","Something went wrong. Please try again later!");
              return mv;
            }

        }else if(req.getParameter("action").equals("Reject")){
            String reject_comment = req.getParameter("reject_comment");
```

```
        if(reject_comment == null || reject_comment == "" || reject_comment.trim().length() < 5 ||
reject_comment.trim().length() > 4000){
            ArrayList<Answer> answerList = approverDao.getAnswerListForApproval();
            mv = new ModelAndView("approve-answers");
            mv.addObject("answerList",answerList);
            mv.addObject("reject",reject_comment);
            mv.addObject("error","Please enter a rejection reason");
            return mv;
        }else{
            long answer_id = Long.parseLong(req.getParameter("answer_id"));
            Answer result = approverDao.approveAnswers(answer_id, approver,reject_comment);
            ArrayList<Answer> answerList = approverDao.getAnswerListForApproval();
            mv = new ModelAndView("approve-answers");
            mv.addObject("answerList",answerList);
            if(result != null){
                 String configEmail = "email-bean.xml";
                ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(configEmail);
                // @Service call
                ApproveEmail approveEmail = (ApproveEmail) context.getBean("approveEmail");
                String toAddr = result.getUser().getEmail();
                String fromAddr = "dedhiadimpi@gmail.com";
                // email subject
                String subject = "Bad news from QuestonsAreWelcome!";
                // email body
                String body = "Hi "+result.getUser().getFirstname()+" "+result.getUser().getLastname()+",\n\nYour Answer
'"+result.getAnswer_description()+"' for Question '"+result.getQuestion().getQuestion_title()+"' is Rejected with
following reason\n"+result.getReject_comment();
                approveEmail.sendEmail(toAddr, fromAddr, subject, body);
                mv.addObject("reject","");
                mv.addObject("error","Answer Rejected successfully!");
           return mv;
            }else{
                mv.addObject("reject",reject_comment);
                mv.addObject("error","Something went wrong. Please try again later!");
                return mv;
            }
        }

      }
      }

      return mv;
   }

}
```

4) User Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controllers.UserController;
```

```
import Dao.AnswerDao.AnswerDao;
import Dao.ApproverDao.ApproverDao;
import Dao.DomainDao.DomainDao;
import Dao.LikesDao.LikesDao;
import Dao.QuestionDao.QuestionDao;
import Models.Answer.Answer;
import Models.Approver.Approver;
import Models.Domain.Domain;
import Models.Likes.Likes;
import Models.Question.Question;
import Models.User.User;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;

/**
 *
 * @author dedhi
 */
public class UserController extends AbstractController{

    @Override
    protected ModelAndView handleRequestInternal(HttpServletRequest request, HttpServletResponse response) throws
Exception {
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        String option = request.getParameter("option");
        if(option.equals("addQuestion")){
            Question question = new Question();
            QuestionDao questionDao = new QuestionDao();
            DomainDao domainDao = new DomainDao();
            String title = request.getParameter("title").trim();
            String desc = request.getParameter("desc").trim();
            List<Question> qList = questionDao.checkQuestion(title,desc);
            if(qList.size() > 0){
                ArrayList<Domain> domainList = domainDao.getDomainList();
                mv = new ModelAndView("add-question");
                mv.addObject("domainList",domainList);
                mv.addObject("title",title);
                mv.addObject("desc",desc);
                mv.addObject("error","Question already exist!");
                return mv;
```

```
      }
      if(title.length() < 5 || title.length() > 500 || desc.length() < 10 || desc.length() > 4000){
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("add-question");
        mv.addObject("domainList",domainList);
        mv.addObject("title",title);
        mv.addObject("desc",desc);
        mv.addObject("error","Title cannot be less than 5 and more than 500 characters! Description cannot be less
than 10 or more than 4000 characters!");
        return mv;
      }
      User user = (User) request.getSession().getAttribute("user");
      question.setQuestion_title(title.trim());
      question.setQuestion_description(desc.trim());
      question.setDomain(domainDao.getDomain(Integer.parseInt(request.getParameter("domain"))));
      question.setUser(user);
      ApproverDao approverDao = new ApproverDao();
      question.setApprover(approverDao.getDummyApprover());
      question.setNo_of_likes(0);
      question.setNo_of_dislikes(0);
      question.setCreated_on(new Date());
      int result = questionDao.addQuestion(question);
      if(result == 1){
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("add-question");
        mv.addObject("domainList",domainList);
        mv.addObject("title","");
        mv.addObject("desc","");
        mv.addObject("error","Your Question is sent for approval. You will receive an email once it's approved!");
        return mv;
      }else{
       ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("add-question");
        mv.addObject("domainList",domainList);
        mv.addObject("title",title);
        mv.addObject("desc",desc);
        mv.addObject("error","Something went wrong. Please try again later!");
        return mv;
      }
    }else if(option.equals("addAnswer")){
      int question_id =Integer.parseInt(request.getParameter("question_id"));
      QuestionDao questionDao = new QuestionDao();
      AnswerDao answerDao = new AnswerDao();
      Question question = questionDao.getQuestion(question_id);
      String desc = request.getParameter("desc").trim();
      List<Answer> aList = answerDao.checkAnswer(request.getParameter("desc"));
      if(aList.size() >0){
        mv = new ModelAndView("add-answer");
        mv.addObject("question", question);
        mv.addObject("desc",desc);
        mv.addObject("error","Answer already exist!");
        return mv;
      }
      if(desc.length() < 10 || desc.length() > 4000){
        mv = new ModelAndView("add-answer");
        mv.addObject("question", question);
        mv.addObject("desc",desc);
        mv.addObject("error","Description cannot be less than 10 or more than 4000 characters!");
        return mv;
      }
      Answer answer = new Answer();
      answer.setAnswer_description(request.getParameter("desc"));
      answer.setUser((User) request.getSession().getAttribute("user"));
      ApproverDao approverDao = new ApproverDao();
      answer.setApprover(approverDao.getDummyApprover());
      answer.setQuestion(question);
      answer.setNo_of_likes(0);
      answer.setNo_of_dislikes(0);
      answer.setCreated_on(new Date());

      int result = answerDao.addAnswer(answer);
      if(result == 1){
        mv = new ModelAndView("add-answer");
        mv.addObject("question", question);
        mv.addObject("desc","");
        mv.addObject("error","Your Answer is sent for approval. You will receive an email once its approved!");
        return mv;
      }else{
        mv = new ModelAndView("add-answer");
        mv.addObject("question", question);
        mv.addObject("desc",desc);
        mv.addObject("error","Something went wrong. Please try again later!");
        return mv;
      }
    }else if(option.equals("editAnswer")){
      AnswerDao answerDao = new AnswerDao();
      int r = 0;
      String desc = request.getParameter("desc").trim();
      List<Answer> aList = answerDao.checkAnswer(desc);
      for(Answer a: aList){
        if(a.getAnswer_id() != Long.parseLong(request.getParameter("answer_id"))){
            r = 1;
            break;
        }
      }
      if(r == 1){
        Answer answer = answerDao.getAnswer(Long.parseLong(request.getParameter("answer_id")));
        mv = new ModelAndView("edit-answer");
        mv.addObject("answer_id",Long.parseLong(request.getParameter("answer_id")));
        mv.addObject("answer_desc",answer.getAnswer_description());
        mv.addObject("error","Answer already exist!");
        return mv;
      }
      if(desc.length() < 10 || desc.length() > 4000){
        Answer answer = answerDao.getAnswer(Long.parseLong(request.getParameter("answer_id")));
        mv = new ModelAndView("edit-answer");
        mv.addObject("answer_id",Long.parseLong(request.getParameter("answer_id")));
        mv.addObject("answer_desc",answer.getAnswer_description());
        mv.addObject("error","Description cannot be less than 10 and greater than 4000 characters!");
        return mv;
      }
      int result = answerDao.updateAnswer(Long.parseLong(request.getParameter("answer_id")),
request.getParameter("desc"));
      User user = (User) request.getSession().getAttribute("user");
      List<Answer> answers = answerDao.getMyAnswerList(user.getUser_id());
      HashMap<Question, ArrayList<Answer>> answerList = new HashMap<>();
      for(Answer a : answers){
        if(answerList.containsKey(a.getQuestion())){
          answerList.get(a.getQuestion()).add(a);
        }else{
          ArrayList<Answer> ans = new ArrayList<Answer>();
          ans.add(a);
          answerList.put(a.getQuestion(), ans);
        }
      }
      mv = new ModelAndView("view-my-answers");
      mv.addObject("answerList",answerList);
      if(result == 1){
        mv.addObject("error","Your answer is sent for approval. You will receive an email once approved!");
      }else{
       mv.addObject("error","Something went wrong. Please try again later!");
      }

      return mv;

    }else if(option.equals("editMyQuestion")){
      String title = request.getParameter("title").trim();
      String desc = request.getParameter("desc").trim();
      QuestionDao questionDao = new QuestionDao();
      List<Question> qList = questionDao.checkQuestion(title,desc);
      int r = 0;
      for(Question q: qList){
        if(q.getQuestion_id() != Long.parseLong(request.getParameter("question_id"))){
          r = 1;
          break;
        }
      }
      if( r == 1){
        User user = (User) request.getSession().getAttribute("user");
        Question question = questionDao.getQuestion(Long.parseLong(request.getParameter("question_id")));
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("edit-question");
        mv.addObject("question", question);
        mv.addObject("domainList", domainList);
        mv.addObject("error","Question already exist!");
        return mv;
      }
      if(title.length() < 5 || title.length() > 500 || desc.length() < 10 || desc.length() > 4000){
```

```
                Question question = questionDao.getQuestion(Long.parseLong(request.getParameter("question_id")));
                DomainDao domainDao = new DomainDao();
                ArrayList<Domain> domainList = domainDao.getDomainList();
                mv = new ModelAndView("edit-question");
                mv.addObject("question", question);
                mv.addObject("domainList", domainList);
                mv.addObject("error","Title cannot be less than 5 and more than 500 characters! Description cannot be less
than 10 or more than 4000 characters");
                return mv;
            }
            int result =
questionDao.updateQuestion(Long.parseLong(request.getParameter("question_id")),request.getParameter("title")
            , request.getParameter("desc"), Long.parseLong(request.getParameter("domain")));
            if(result == 1){
                User user = (User) request.getSession().getAttribute("user");
                List<Question> questionList = questionDao.getMyQuestionList(user.getUser_id());
                DomainDao domainDao = new DomainDao();
                ArrayList<Domain> domainList = domainDao.getDomainList();
                mv = new ModelAndView("view-all-questions");
                mv.addObject("questionList", questionList);
                mv.addObject("action", "myQuestions");
                mv.addObject("domainList",domainList);
//                  ApproverDao approverDao = new ApproverDao();
//                  Approver approver = approverDao.getDummyApprover();
//                  mv.addObject("approver_id",approver.getApprover_id());
                mv.addObject("error","Your Question is sent for approval. You will receive an email once it's approved!");
                return mv;
            }else{
                User user = (User) request.getSession().getAttribute("user");
                Question question = questionDao.getQuestion(Long.parseLong(request.getParameter("question_id")));
                DomainDao domainDao = new DomainDao();
                ArrayList<Domain> domainList = domainDao.getDomainList();
                mv = new ModelAndView("edit-question");
                mv.addObject("question", question);
                mv.addObject("domainList", domainList);
                mv.addObject("error","Something went wrong. Please try again later!");
                return mv;
            }

        }else{
            String select = request.getParameter("option");
            User user = (User) request.getSession().getAttribute("user");
            QuestionDao questionDao = new QuestionDao();
            LikesDao likesDao = new LikesDao();
            if(select.equals("like_question")){
                int result = 0;
                long question_id = Long.parseLong(request.getParameter("question_id"));
                Question question = questionDao.getQuestion(question_id);
                Likes like = likesDao.getLikeQuestion(question_id, user.getUser_id());
                if(like == null){
                    like = new Likes();
                    like.setQuestion(question);
                    like.setUser(user);
                    like.setType("like");
                    int likes = question.getNo_of_likes();
                    question.setNo_of_likes(likes+1);
                    result = questionDao.addLikes(question);
                }else if(like.getType().equals("dislike")){
                    like.setType("like");
                    int dislikes = question.getNo_of_dislikes();
                    question.setNo_of_dislikes(dislikes-1);
                    int likes = question.getNo_of_likes();
                    question.setNo_of_likes(likes+1);
                    result = questionDao.addLikes(question);
                }
                result = likesDao.addLikes(like);
                if(result == 1){
                    List<Question> questionList = questionDao.getQuestionList();
                    mv = new ModelAndView("view-all-questions");
                    mv.addObject("questionList", questionList);
                    mv.addObject("action", "allQuestions");
                    return mv;
                }else{
                    mv = new ModelAndView("login-fail");
                    return mv;
                }
            }else if(select.equals("dislike_question")){
                int result = 0;
                long question_id = Long.parseLong(request.getParameter("question_id"));
                Question question = questionDao.getQuestion(question_id);
                Likes like = likesDao.getLikeQuestion(question_id, user.getUser_id());
                if(like == null){
                    like = new Likes();
                    like.setQuestion(question);
                    like.setUser(user);
                    like.setType("dislike");
                    int dislikes = question.getNo_of_dislikes();
                    question.setNo_of_dislikes(dislikes+1);
                    result = questionDao.addLikes(question);
                }else if(like.getType().equals("like")){
                    like.setType("dislike");
                    int dislikes = question.getNo_of_dislikes();
                    question.setNo_of_dislikes(dislikes+1);
                    int likes = question.getNo_of_likes();
                    question.setNo_of_likes(likes-1);
                    result = questionDao.addLikes(question);
                }
                result = likesDao.addLikes(like);
                if(result == 1){
                    List<Question> questionList = questionDao.getQuestionList();
                    mv = new ModelAndView("view-all-questions");
                    mv.addObject("questionList", questionList);
                    mv.addObject("action", "allQuestions");
                    return mv;
                }else{
                    mv = new ModelAndView("login-fail");
            return mv;
         }

}else if(select.equals("like_answer")){
   int result = 0;
   long answer_id = Long.parseLong(request.getParameter("answer_id"));
   AnswerDao answerDao = new AnswerDao();
   Answer answer = answerDao.getAnswer(answer_id);
   Likes like = likesDao.getLikeAnswer(answer_id, user.getUser_id());
   if(like == null){
      like = new Likes();
      like.setAnswer(answer);
      like.setUser(user);
      like.setType("like");
      int likes = answer.getNo_of_likes();
      answer.setNo_of_likes(likes+1);
      result = answerDao.addLikes(answer);

   }else if(like.getType().equals("dislike")){
      like.setType("like");
      int dislikes = answer.getNo_of_dislikes();
      answer.setNo_of_dislikes(dislikes-1);
      int likes = answer.getNo_of_likes();
      answer.setNo_of_likes(likes+1);
      result = answerDao.addLikes(answer);
   }
   result = likesDao.addLikes(like);
   if(result == 1){
      List<Question> questionList = questionDao.getQuestionList();
      mv = new ModelAndView("view-all-questions");
      mv.addObject("questionList", questionList);
      mv.addObject("action", "allQuestions");
      return mv;
   }else{
      mv = new ModelAndView("login-fail");
      return mv;
   }

}else if(select.equals("dislike_answer")){
   int result = 0;
   long answer_id = Long.parseLong(request.getParameter("answer_id"));
   AnswerDao answerDao = new AnswerDao();
   Answer answer = answerDao.getAnswer(answer_id);
   Likes like = likesDao.getLikeAnswer(answer_id, user.getUser_id());
   if(like == null){
      like = new Likes();
      like.setAnswer(answer);
      like.setUser(user);
      like.setType("dislike");
      int dislikes = answer.getNo_of_dislikes();
      answer.setNo_of_dislikes(dislikes+1);
      result = answerDao.addLikes(answer);
```

```
                }else if(like.getType().equals("like")){
                    like.setType("dislike");
                    int dislikes = answer.getNo_of_dislikes();
                    answer.setNo_of_dislikes(dislikes+1);
                    int likes = answer.getNo_of_likes();
                    answer.setNo_of_likes(likes-1);
                    result = answerDao.addLikes(answer);
                }
                result = likesDao.addLikes(like);

                if(result == 1){
                    List<Question> questionList = questionDao.getQuestionList();
                    mv = new ModelAndView("view-all-questions");
                    mv.addObject("questionList", questionList);
                    mv.addObject("action", "allQuestions");
                    return mv;
                }else{
                    mv = new ModelAndView("login-fail");
                    return mv;
                }
            }
        }
        return mv;
    }

}
```

5) Redirect Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controllers.RedirectController;

import Dao.AnswerDao.AnswerDao;
import Dao.ApproverDao.ApproverDao;
import Dao.DomainDao.DomainDao;
import LoginDao.LoginDao;
import Dao.QuestionDao.QuestionDao;
import Dao.UserDao.UserDao;
import Models.Answer.Answer;
import Models.Approver.Approver;
import Models.Domain.Domain;
import Models.Question.Question;
import Models.User.User;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import Helper.Validation;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;

/**
 *
 * @author dedhi
 */
@Controller
public class RedirectController{
//
//    @Override
//    protected ModelAndView handleRequestInternal(HttpServletRequest request, HttpServletResponse response)
throws Exception {
//        ModelAndView mv = null;
//        mv = new ModelAndView("add-question");
//        return mv;
//    }

    @RequestMapping(value = "/redirect-add-question.htm")
    protected ModelAndView getAddQuestionModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("add-question");
        mv.addObject("domainList",domainList);
        mv.addObject("error","");
        mv.addObject("title","");
        mv.addObject("desc","");
        return mv;
    }

    @RequestMapping(value = "/redirect-user-home.htm")
    protected ModelAndView getUserHomeModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        mv = new ModelAndView("user-home");
        return mv;
    }
    @RequestMapping(value = "/redirect-admin-home.htm")
    protected ModelAndView getAdminHomeModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        mv = new ModelAndView("admin-home");
        return mv;
    }

    @RequestMapping(value = "/redirect-approver-home.htm")
    protected ModelAndView getApproverHomeModel(HttpServletRequest request, HttpServletResponse response)
throws Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        mv = new ModelAndView("approver-home");
        return mv;
    }

    @RequestMapping(value="/redirect-get-question-list.htm")
    protected ModelAndView getQuestionList(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        QuestionDao questionDao = new QuestionDao();
        List<Question> questionList = questionDao.getQuestionList();
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("view-all-questions");
        mv.addObject("questionList", questionList);
        mv.addObject("action", "allQuestions");
        mv.addObject("domainList",domainList);
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value="/redirect-get-approver-question-list.htm")
    protected ModelAndView getApproverQuestionList(HttpServletRequest request, HttpServletResponse response)
throws Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        QuestionDao questionDao = new QuestionDao();
        List<Question> questionList = questionDao.getApproverQuestionList();
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("view-all-questions");
        mv.addObject("questionList", questionList);
        mv.addObject("action", "approverQuestions");
        mv.addObject("domainList",domainList);
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value="/redirect-get-my-question-list.htm")
    protected ModelAndView getMyQuestionList(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        User user = (User) request.getSession().getAttribute("user");
        QuestionDao questionDao = new QuestionDao();
        List<Question> questionList = questionDao.getMyQuestionList(user.getUser_id());
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("view-all-questions");
        mv.addObject("questionList", questionList);
        mv.addObject("action", "myQuestions");
        mv.addObject("domainList",domainList);
//        ApproverDao approverDao = new ApproverDao();
//        Approver approver = approverDao.getDummyApprover();
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value="/redirect-search-list.htm")
    protected ModelAndView getSearchQuestionList(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        long domain_id = Long.parseLong(request.getParameter("domain"));
        QuestionDao questionDao = new QuestionDao();
        List<Question> questionList = questionDao.searchQuestionList(request.getParameter("keyword"), domain_id);
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        if(request.getParameter("action").equals("allQuestions")){
            mv = new ModelAndView("view-all-questions");
            mv.addObject("questionList", questionList);
            mv.addObject("action", "allQuestions");
            mv.addObject("domainList",domainList);
            mv.addObject("error","");
        return mv;
        }else if(request.getParameter("action").equals("myQuestions")){
            mv = new ModelAndView("view-all-questions");
            mv.addObject("questionList", questionList);
            mv.addObject("action", "myQuestions");
            mv.addObject("domainList",domainList);
            mv.addObject("error","");
        return mv;

        }else if(request.getParameter("action").equals("approverQuestions")){
            List<Question> questions = questionDao.searchApproverQuestionList(request.getParameter("keyword"),
domain_id);
            mv = new ModelAndView("view-all-questions");
            mv.addObject("questionList", questions);
            mv.addObject("action", "approverQuestions");
            mv.addObject("domainList",domainList);
            mv.addObject("error","");
            return mv;
        }
        return mv;

    }

    @RequestMapping(value="/redirect-edit-question.htm")
    protected ModelAndView editMyQuestion(@RequestParam("question_id") long question_id, HttpServletRequest
request, HttpServletResponse response) throws Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        User user = (User) request.getSession().getAttribute("user");
        QuestionDao questionDao = new QuestionDao();
        Question question = questionDao.getQuestion(question_id);
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("edit-question");
        mv.addObject("question", question);
        mv.addObject("domainList", domainList);
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value="/redirect-get-my-answer-list.htm")
    protected ModelAndView getMyAnswerList(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        User user = (User) request.getSession().getAttribute("user");
        AnswerDao answerDao = new AnswerDao();
        List<Answer> answers = answerDao.getMyAnswerList(user.getUser_id());
        HashMap<Question, ArrayList<Answer>> answerList = new HashMap<>();
        for(Answer a : answers){
            if(answerList.containsKey(a.getQuestion())){
                answerList.get(a.getQuestion()).add(a);
            }else{
                ArrayList<Answer> ans = new ArrayList<Answer>();
                ans.add(a);
                answerList.put(a.getQuestion(), ans);
            }
        }
        mv = new ModelAndView("view-my-answers");
        mv.addObject("answerList",answerList);
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value = "redirect-add-answer.htm")
    protected ModelAndView getAddAnswerModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        QuestionDao questionDao = new QuestionDao();
        long id = Long.parseLong(request.getParameter("id"));
        Question question = questionDao.getQuestion(id);
        mv = new ModelAndView("add-answer");
        mv.addObject("question", question);
        mv.addObject("desc","");
        mv.addObject("error","");
        return mv;
    }

    @RequestMapping(value = "redirect-my-profile.htm")
    protected ModelAndView getMyProfileModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        String role = (String) request.getSession().getAttribute("role");
        if(role.equals("user")){
            User myProfile = (User) request.getSession().getAttribute("user");
            mv = new ModelAndView("my-profile");
            mv.addObject("myProfile", myProfile);
            mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
            mv.addObject("disabled","disabled");
            mv.addObject("view","true");
            mv.addObject("error","");
        }else if(role.equals("approver")){
            Approver myProfile = (Approver) request.getSession().getAttribute("user");
            mv = new ModelAndView("my-profile");
            mv.addObject("myProfile", myProfile);
            mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
            mv.addObject("disabled","disabled");
            mv.addObject("view","true");
            mv.addObject("error","");
        }
        return mv;
    }

    @RequestMapping(value = "redirect-edit-myProfile.htm")
    protected ModelAndView editMyProfileModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        String role = (String) request.getSession().getAttribute("role");
        if(role.equals("user")){
            User myProfile = (User) request.getSession().getAttribute("user");
            mv = new ModelAndView("my-profile");
            mv.addObject("myProfile", myProfile);
            mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
            mv.addObject("disabled","");
            mv.addObject("view","false");
            mv.addObject("error","");
        }else if(role.equals("approver")){
            Approver myProfile = (Approver) request.getSession().getAttribute("user");
            mv = new ModelAndView("my-profile");
            mv.addObject("myProfile", myProfile);
            mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
            mv.addObject("disabled","");
            mv.addObject("view","false");
            mv.addObject("error","");
        }

        return mv;
    }

    @RequestMapping(value = "redirect-save-myProfile.htm")
    protected ModelAndView saveMyProfileModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        Validation validation = new Validation();
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        long user_id = Long.parseLong(request.getParameter("user_id"));
        String firstname = request.getParameter("firstname");
        String lastname = request.getParameter("lastname");
        String gender = request.getParameter("gender");
        Date date = new Date();
        String address = request.getParameter("address");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        int result = 0;
        if(request.getSession().getAttribute("role").equals("user")){
            User myProfile = (User) request.getSession().getAttribute("user");
            try{
              date = new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("dateofbirth").substring(0, 9));
            }catch(Exception e){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter correct date!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            if(date.compareTo(new Date()) > 0){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter a date less than today's date!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            if(gender.equals("")){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter your Gender!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.onlyLettersValidation(firstname) || firstname.trim().length() < 0 || firstname.trim().length() >
50){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid First Name! First name cannot have more than 50 letters");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.onlyLettersValidation(lastname) || lastname.trim().length() < 0 || lastname.trim().length() >
50){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Last Name! Last name cannot have more than 50 letters");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(address.trim().length() < 8 || address.trim().length() > 100){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Address! Address cannot be less than 8 letters or more than 100
letters!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.phoneNumberValidation(phone) || phone.trim().length() > 20){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Phone Number!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            UserDao userDao = new UserDao();
            result = userDao.updateUser(user_id, firstname, lastname, gender, date, address, email, phone);
          }else if(request.getSession().getAttribute("role").equals("approver")){
            Approver myProfile = (Approver) request.getSession().getAttribute("user");
            try{
              date = new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("dateofbirth").substring(0, 9));
            }catch(Exception e){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter correct date!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            if(date.compareTo(new Date()) > 0){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter a date less than today's date!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            if(gender.equals("")){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter your Gender!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.onlyLettersValidation(firstname) || firstname.trim().length() < 0 || firstname.trim().length() >
      50){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid First Name! First name cannot have more than 50 letters");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.onlyLettersValidation(lastname) || lastname.trim().length() < 0 || lastname.trim().length() >
      50){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Last Name! Last name cannot have more than 50 letters");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(address.trim().length() < 8 || address.trim().length() > 100){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Address! Address cannot be less than 8 letters or more than 100
      letters!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }else if(!validation.phoneNumberValidation(phone) || phone.trim().length() > 20){
              mv = new ModelAndView("my-profile");
              mv.addObject("error","Please enter valid Phone Number!");
              mv.addObject("myProfile", myProfile);
              mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
              mv.addObject("disabled","");
              mv.addObject("view","false");
              return mv;
            }
            ApproverDao approverDao = new ApproverDao();
            result = approverDao.updateApprover(user_id, firstname, lastname, gender, date, address, email, phone);
          }
            String role = (String) request.getSession().getAttribute("role");
            if(role.equals("user")){
              User myProfile = (User) request.getSession().getAttribute("user");

              if(result == 1){
                myProfile.setFirstname(firstname);
                myProfile.setLastname(lastname);
                myProfile.setGender(gender);
                myProfile.setDate_of_birth(date);
                myProfile.setAddress(address);
                myProfile.setEmail(email);
                myProfile.setPhone_number(phone);
              }
              mv = new ModelAndView("my-profile");
```

```
                mv.addObject("myProfile", myProfile);
                mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
                mv.addObject("disabled","disabled");
                mv.addObject("view","true");
                if(result == 1){
                    mv.addObject("error","Profile updated successfully!");
                }else{
                    mv.addObject("error","Something went wrong. Please try again later!");
                }
            }else if(role.equals("approver")){
                Approver myProfile = (Approver) request.getSession().getAttribute("user");
                if(result == 1){
                    myProfile.setFirstname(firstname);
                    myProfile.setLastname(lastname);
                    myProfile.setGender(gender);
                    myProfile.setDate_of_birth(date);
                    myProfile.setAddress(address);
                    myProfile.setEmail(email);
                    myProfile.setPhone_number(phone);
                }
                mv = new ModelAndView("my-profile");
                mv.addObject("myProfile", myProfile);
                mv.addObject("date",new SimpleDateFormat("yyyy-MM-dd").format(myProfile.getDate_of_birth()));
                mv.addObject("disabled","disabled");
                mv.addObject("view","true");
                if(result == 1){
                    mv.addObject("error","Profile updated successfully!");
                }else{
                    mv.addObject("error","Something went wrong. Please try again later!");
                }
            }
            return mv;
    }

    @RequestMapping(value = "redirect-edit-answer.htm")
    protected ModelAndView editAnswerModel(@RequestParam("answer_id") long answer_id, HttpServletRequest
request, HttpServletResponse response) throws Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
        AnswerDao answerDao = new AnswerDao();
            Answer answer = answerDao.getAnswer(answer_id);
            mv = new ModelAndView("edit-answer");
            mv.addObject("answer_id",answer_id);
            mv.addObject("answer_desc",answer.getAnswer_description());
            mv.addObject("error","");
            return mv;
        }

    @RequestMapping(value = "redirect-delete-answer.htm")
```

```
  protected ModelAndView deleteAnswerModel(@RequestParam("answer_id") long answer_id, HttpServletRequest
request, HttpServletResponse response) throws Exception{
      ModelAndView mv = null;
      if(request.getSession().getAttribute("user") == null){
      mv = new ModelAndView("login");
      return mv;
    }
      AnswerDao answerDao = new AnswerDao();
      int result = answerDao.deleteAnswer(answer_id);
      User user = (User) request.getSession().getAttribute("user");
        List<Answer> answers = answerDao.getMyAnswerList(user.getUser_id());
        HashMap<Question, ArrayList<Answer>> answerList = new HashMap<>();
        for(Answer a : answers){
          if(answerList.containsKey(a.getQuestion())){
            answerList.get(a.getQuestion()).add(a);
          }else{
            ArrayList<Answer> ans = new ArrayList<Answer>();
            ans.add(a);
            answerList.put(a.getQuestion(), ans);
          }
        }
        mv = new ModelAndView("view-my-answers");
        mv.addObject("answerList",answerList);

      if(result == 1){
        mv.addObject("error","Answer deleted successfully!");
      }else{
        mv.addObject("error","Something went wrong. Please try again later!");
      }
      return mv;
    }

  @RequestMapping(value = "redirect-delete-question.htm")
  protected ModelAndView deleteQuestionModel(@RequestParam("question_id") long question_id,
HttpServletRequest request, HttpServletResponse response) throws Exception{
      ModelAndView mv = null;
      if(request.getSession().getAttribute("user") == null){
      mv = new ModelAndView("login");
      return mv;
    }
      QuestionDao questionDao = new QuestionDao();
      int result = questionDao.deleteQuestion(question_id);
      if(result == 1){
        User user = (User) request.getSession().getAttribute("user");
        List<Question> questionList = questionDao.getMyQuestionList(user.getUser_id());
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("view-all-questions");
        mv.addObject("questionList", questionList);
        mv.addObject("action", "myQuestions");
        mv.addObject("domainList",domainList);
        mv.addObject("error","Question deleted successfully!");
        return mv;
      }else{
        User user = (User) request.getSession().getAttribute("user");
        List<Question> questionList = questionDao.getMyQuestionList(user.getUser_id());
        DomainDao domainDao = new DomainDao();
        ArrayList<Domain> domainList = domainDao.getDomainList();
        mv = new ModelAndView("view-all-questions");
        mv.addObject("questionList", questionList);
        mv.addObject("action", "myQuestions");
        mv.addObject("domainList",domainList);
        mv.addObject("error","Something went wrong. Please try again later!");
        return mv;
      }
    }

  @RequestMapping(value = "redirect-approve-questions.htm")
  protected ModelAndView approveQuestionsModel(HttpServletRequest request, HttpServletResponse response)
throws Exception{
    ModelAndView mv = null;
    if(request.getSession().getAttribute("user") == null){
      mv = new ModelAndView("login");
      return mv;
    }
      ApproverDao approverDao = new ApproverDao();
      ArrayList<Question> questionList = approverDao.getQuestionListForApproval();
      mv = new ModelAndView("approve-questions");
      mv.addObject("questionList",questionList);
      mv.addObject("reject","");
      if(questionList.isEmpty()){
        mv.addObject("error","No pending approvals");
      }else{
        mv.addObject("error","");
      }

      return mv;
    }

  @RequestMapping(value = "redirect-approve-answers.htm")
  protected ModelAndView approveAnswersModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
    ModelAndView mv = null;
    if(request.getSession().getAttribute("user") == null){
      mv = new ModelAndView("login");
      return mv;
    }
      ApproverDao approverDao = new ApproverDao();
      ArrayList<Answer> answerList = approverDao.getAnswerListForApproval();
      mv = new ModelAndView("approve-answers");
      mv.addObject("answerList",answerList);
      mv.addObject("reject","");
      if(answerList.isEmpty()){
        mv.addObject("error","No pending approvals");
      }else{
        mv.addObject("error","");
```

```
            }
            return mv;
        }

    @RequestMapping(value = "redirect-register-user.htm")
    protected ModelAndView registerUserModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
            mv = new ModelAndView("register-user");
            mv.addObject("firstname","");
            mv.addObject("lastname","");
            mv.addObject("date", new Date());
            mv.addObject("address","");
            mv.addObject("email","");
            mv.addObject("phone","");
            mv.addObject("username","");
            mv.addObject("password","");
            mv.addObject("error","");
            return mv;
        }

    @RequestMapping(value = "redirect-add-approver.htm")
    protected ModelAndView addApproverModel(HttpServletRequest request, HttpServletResponse response) throws
Exception{
        ModelAndView mv = null;
        if(request.getSession().getAttribute("user") == null){
            mv = new ModelAndView("login");
            return mv;
        }
            mv = new ModelAndView("add-approver");
            mv.addObject("firstname","");
            mv.addObject("lastname","");
            mv.addObject("date", new Date());
            mv.addObject("address","");
            mv.addObject("email","");
            mv.addObject("phone","");
            mv.addObject("username","");
            mv.addObject("password","");
            mv.addObject("error","");
            return mv;
        }

}
```