**A**

**Project Report on**

**SANGOLA MAHAVIDYALAYA,SANGOLA**

**PROJECT REPORT**

**ON**

# “Railway Reservation System”

**Submitted In Partial Fulfillment Of The**

**Requirement of The**

**Post-Graduation Course Of**

**M.Sc. Computer Science-II**

**2023- 2024**

**SUBMITTED BY**

# MISS. KAJAL BHAUSAHEB SHINDE

**Of**

**Punyashlok Ahilyadevi Holkar Solapur University**

Guided By,

**Prof. Tathe sir**

**Punyashlok Ahilyadevi Holkar Solapur**

**University,Solapur**



**Department of Computer Science**

*Seat No. : -………….**Date: …………….*

*This is to certify that* ***Miss. Kajal Bhausaheb Shinde****. Has*

*Contributed in successfully completing the project titled* ***“Railway Reservation System”*** *in satisfactory manner as a partial fulfillment for the course of* ***MSc (Computer Science)-II*** *for the session 2023-2024 of the University of Solapur.*

# *\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***[Project Guide] [Head of Department]***

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[External Examiner]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Internal Examiner]**

**INDEX**

* Declaration

* Acknowledgement

* Synopsis

* Requirements

* User Manual

* Scope of the system

* System Flow Diagram

* Source Code

* Input

* Output

* Feasibility Study

* Limitations

* Bibliography

# *DECLARATION*

DECLARATION

## *I declare that we have referred the books given*

***in the bibliography during the project work and have not copied any written material or its part there of this project work is purely created on our own and on my merit. In this project the main feature of the projects is to store the result information of student for some times.We can also able to modify ,sort, search any student inform- ation easily. Also we easily watch the inserted records by specifically.***

***This project work was made possible because of valuable guidance of Project guide Prof .* Tathe sir *also received some valuable tips from other teachers in our Department.***

***Miss. Shinde Kajal Bhausaheb***

# ACKNOWLEDGEMENT

***We would like to add few heartfelt words for the people who gave ending support and for successful completion of the project.***

***We take this opportunity to express our deep sense of gratitude to* Prof. Tathe sir** ***for his invaluable Guidance through out the project.***

***We gave special thanks to* Prof . Tathe sir** ***and all the faculty members of computer Department for their constant encouragement and***  ***valuable guidance for completing this project.***

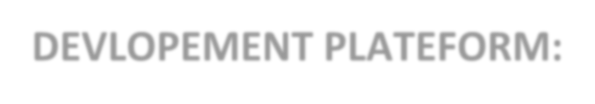
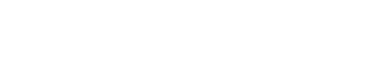
***This has been great experience for us. We have benefited a lot from the Constructive Criticism and***

***Suggestions given to us by all faculty members and management of sangola college.***

***We also thank all our friends , seniors and well-wishers for their co-operation that encouraged us immensely complete this project.***

***Miss.Kajal Bhausaheb Shinde.***

**DEVLOPEMENT PLATEFORM:**



***Operating System*** :- *Windows XP.*

***Platform*** :- Visual studio 2019 ***Language*** :- .Net Framework(C# -Entity framework)



## Requirements

### Hardware Requirements

* Operating Systems- Windows, macOS, Linux, Chrome OS
* Windows 8.0, 8.1 and 10, 11 (32-bit and 64-bit)
* Random Access Memory (RAM) - 1 GB RAM minimum; 8 GB RAM recommended.
* 1.6 GHz or faster processor.
* Internet Connection (20MBps or above)

### Software Requirements

* Microsoft Windows XP or later versions ▪ A standard web browser.
* .Net framework.
* Visual Studio Code above 2014

## SOURCE CODE

HomeController.cs

﻿using RailwayReservationMVC.Models;

using System;

using System.Linq;

using System.Net.Mail;

using System.Net;

using System.Web.Mvc;

using System.Text;

using RailwayReservationMVC.Models.DAL;

using System.Collections.Generic;

namespace RailwayReservationMVC.Controllers

{

public class HomeController : Controller

{

#region Repository Object

private IRailwayRepository<Reservation> ReservationObject;

private IRailwayRepository<TrainDetails> TrainDetailsObject;

private IRailwayRepository<User> UserObject;

#endregion

#region Repository Constructor

public HomeController()

{

this.ReservationObject = new RailwayRepository<Reservation>();

this.TrainDetailsObject = new RailwayRepository<TrainDetails>();

this.UserObject = new RailwayRepository<User>();

}

#endregion

#region Homepage

public ActionResult Homepage()

{

var data = TrainDetailsObject.GetModel().ToList();

return View(data);

}

#endregion

#region Search Train

[HttpPost]

public ActionResult Homepage(string searching)

{

return View(TrainDetailsObject.GetModel().Where(x => x.DestinationStation.Contains(searching) || searching == null).ToList());

}

#endregion

#region SignUp and Email Notification

public ActionResult SignUp()

{

return View();

}

[HttpPost]

public ActionResult SignUp(User u)

{

if(ModelState.IsValid)

{

//email notification

MailMessage mm = new MailMessage("railwayreservationsystemmail@gmail.com", u.Email);

mm.Subject = "Welcome to Railway Reservation System";

mm.Body = "This is your password :" + u.Password.ToString() + "\n" + "Have a safe journey with us !";

mm.IsBodyHtml = false;

SmtpClient smtp = new SmtpClient();

smtp.Host = "smtp.gmail.com";

smtp.Port = 587;

smtp.EnableSsl = true;

NetworkCredential nc = new NetworkCredential("railwayreservationsystemmail@gmail.com", "chfxpbtcfjfobhlv");

smtp.UseDefaultCredentials = true;

smtp.Credentials = nc;

smtp.Send(mm);

TempData["Message"] = "Thank you for Connecting with us!Your password has been sent to your registered Emailid ";

//password encryption

u.Password = Convert.ToBase64String(System.Security.Cryptography.SHA256.Create().ComputeHash(Encoding.UTF8.GetBytes(u.Password)));

//update database with new user

UserObject.InsertModel(u);

UserObject.Save();

//session for userid

Session["user\_id"] = u.User\_Id;

Session["email"] = u.Email;

return RedirectToAction("afterlogin");

}

else

{

TempData["InvalidCredentials"] = "Invalid SignUp Credentials";

return View();

}

}

#endregion

#region Login

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(string email,string password)

{

if (ModelState.IsValid)

{

List<User> userDetails = UserObject.GetModel().ToList();

var userid = from id in userDetails where id.Email ==email select id.User\_Id;

Session["user\_id"] = userid;

//password encryption to validate with database encrypted password

password = Convert.ToBase64String(

System.Security.Cryptography.SHA256.Create().ComputeHash(Encoding.UTF8.GetBytes(password)));

var data = UserObject.GetModel().Where(s => s.Email.Equals(email) && s.Password.Equals(password)).ToList();

if (data.Count() > 0)

{

return RedirectToAction("afterlogin");

}

else

{

ViewBag.error = "Login failed";

return RedirectToAction("Login");

}

}

return View();

}

public ActionResult AdminLogin()

{

return View();

}

[HttpPost]

public ActionResult AdminLogin(string email,string password,string role)

{

if (ModelState.IsValid)

{

var data = UserObject.GetModel().Where(s => s.Email.Equals(email) && s.Password.Equals(password) && s.Role.Equals(role)).ToList();

if (data.Count() > 0)

{

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

}

else

{

ViewBag.error = "Login failed";

return RedirectToAction("Login");

}

}

return View();

}

public ActionResult afterlogin()

{

return View();

}

#endregion

#region logout

public ActionResult Logout()

{

return RedirectToAction("Homepage");

}

#endregion

#region Reservation

public ActionResult ReservationPage()

{

var data = ReservationObject.GetModel();

return View(data);

}

public ActionResult CreateReservation()

{

return View();

}

[HttpPost]

public ActionResult CreateReservation(Reservation res)

{

var data = ReservationObject.GetModel();

if (data.Count() >= 7)

{

TempData["msg"] = "<script>alert('YOU CANNOT ADD MORE THAN 6 PASSENGER');</script>";

}

else

{

ReservationObject.InsertModel(res);

ReservationObject.Save();

return RedirectToAction("ReservationPage");

}

return View(res);

}

public ActionResult DeleteReservation(int id)

{

Reservation rt = ReservationObject.GetModelByID(id);

return View(rt);

}

[HttpPost]

public ActionResult DeleteReservation(int id, Reservation reservation)

{

ReservationObject.DeleteModel(id);

ReservationObject.Save();

return RedirectToAction("ReservationPage");

}

#endregion

#region Search Train

public ActionResult ViewTrainDetails()

{

var data = TrainDetailsObject.GetModel().ToList();

return View(data);

}

[HttpPost]

public ActionResult ViewTrainDetails(string searching)

{

return View(TrainDetailsObject.GetModel().Where(x => x.DestinationStation.Contains(searching) || searching == null).ToList());

}

#endregion

}

}

AdminController.cs

﻿using RailwayReservationMVC.Models;

using RailwayReservationMVC.Models.DAL;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace RailwayReservationMVC.Controllers

{

public class AdminController : Controller

{

#region repositorypattern

private IRailwayRepository<TrainDetails> interfaceObj;

public AdminController()

{

this.interfaceObj = new RailwayRepository<TrainDetails>();

}

#endregion

#region admin crudoperations

public ActionResult Index()

{

var data = interfaceObj.GetModel().ToList();

return View(data);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(TrainDetails train)

{

interfaceObj.InsertModel(train);

interfaceObj.Save();

return RedirectToAction("Index");

//return View();

}

public ActionResult Edit(int id)

{

TrainDetails td = interfaceObj.GetModelByID(id);

return View(td);

}

[HttpPost]

public ActionResult Edit(int id, TrainDetails train)

{

interfaceObj.UpdateModel(train);

interfaceObj.Save();

return RedirectToAction("Index");

}

public ActionResult Delete(int id)

{

TrainDetails td = interfaceObj.GetModelByID(id);

return View(td);

}

[HttpPost]

public ActionResult Delete(int id, TrainDetails train)

{

interfaceObj.DeleteModel(id);

interfaceObj.Save();

return RedirectToAction("Index");

}

}

#endregion

}

PaymentController.cs

﻿using RailwayReservationMVC.Models.DAL;

using RailwayReservationMVC.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Numerics;

using System.Web.Util;

using System.Net.Mail;

using System.Net;

using System.Web.Helpers;

namespace RailwayReservationMVC.Controllers

{

public class PaymentController : Controller

{

public string transactionId;

// GET: Payment

private IRailwayRepository<Reservation> ResObj;

private IRailwayRepository<TrainDetails> TrainObj;

public PaymentController()

{

this.ResObj = new RailwayRepository<Reservation>();

this.TrainObj = new RailwayRepository<TrainDetails>();

}

public ActionResult Index()

{

return View();

}

[HttpPost]

public ActionResult CreateOrder(Models.OrderModel \_requestData)

{

Guid guidtrans = Guid.NewGuid();

BigInteger big = new BigInteger(guidtrans.ToByteArray());

var bigstr = big.ToString();

var Id = bigstr.Replace("-", string.Empty);

var transactionId = Id.ToString().Substring(0, 12);

Session["transactionId"] = transactionId;

Razorpay.Api.RazorpayClient client = new Razorpay.Api.RazorpayClient("rzp\_test\_aJVAJtaSskH7US", "5ug92ZJIU89n0R3SKU7O0RP8");

Dictionary<string, object> options = new Dictionary<string, object>();

options.Add("amount", 500 \* 100);

options.Add("receipt", transactionId);

options.Add("currency", "INR");

options.Add("payment\_capture", "0");

Razorpay.Api.Order orderResponse = client.Order.Create(options);

string orderId = orderResponse["id"].ToString();

Models.OrderModel orderModel = new Models.OrderModel

{

orderId = orderResponse.Attributes["id"],

razorpayKey = "rzp\_test\_aJVAJtaSskH7US",

amount = 500 \* 100,

currency = "INR",

name = \_requestData.name,

email = \_requestData.email,

contactNumber = \_requestData.contactNumber,

address = \_requestData.address,

};

return View("PaymentPage", orderModel);

}

[HttpPost]

public ActionResult Complete(Reservation reserve)

{

string paymentId = Request.Params["rzp\_paymentid"];

string orderId = Request.Params["rzp\_orderid"];

Razorpay.Api.RazorpayClient client = new Razorpay.Api.RazorpayClient("rzp\_test\_aJVAJtaSskH7US", "5ug92ZJIU89n0R3SKU7O0RP8");

Razorpay.Api.Payment payment = client.Payment.Fetch(paymentId);

Dictionary<string, object> options = new Dictionary<string, object>();

options.Add("amount", payment.Attributes["amount"]);

Razorpay.Api.Payment paymentCaptured = payment.Capture(options);

string amt = paymentCaptured.Attributes["amount"];

//Session["email\_to"] = \_requestData.email;

if (paymentCaptured.Attributes["status"] == "captured")

{

//sessions

Guid guid = Guid.NewGuid();

BigInteger big = new BigInteger(guid.ToByteArray());

var bigstr = big.ToString();

var str = bigstr.Replace("-", string.Empty);

var pnr = str.ToString().Substring(0, 10);

Session["PNR"] = pnr;

List<TrainDetails> trainDetails = TrainObj.GetModel().ToList();

reserve.PNR\_NO = pnr;

reserve.Res\_Name = (string)Session["ResName"];

reserve.Res\_Gender = (string)Session["ResGender"];

reserve.QuotaType= (string)Session["Quota"];

reserve.User\_Id = (int)Session["user\_id"];

reserve.Res\_Date = (string)Session["Date"];

reserve.Seat\_No = (int)Session["seatno"];

//var train\_Id = from id in trainDetails where id.SourceStation == reserve.TrainDetails.SourceStation select id;

reserve.Transaction\_Id = (string)Session["transactionId"];

var train\_id = from id in trainDetails where id.SourceStation == (string)Session["SourceStation"] select id.Train\_Id;

var tf = train\_id.FirstOrDefault();

reserve.Train\_Id = tf;

ResObj.InsertModel(reserve);

ResObj.Save();

//email notification:

MailMessage mm = new MailMessage("railwayreservationsystemmail@gmail.com", (string)Session["email"] );

mm.Subject = "Welcome to Railway Reservation System";

mm.Body = "Thank you for Booking tickets with us!" + "\n" + "Here are you details:" + "\n" + "Transaction Id :" + Session["transactionId"] + "\n" + "PNR Number" + Session["PNR"] + "\n"+ "Name" + Session["ResName"] +"\n"+ "Source Station" + Session["SourceStation"] + "\n" + "Destination Station" + Session["DestinationStation"] + "\n" + "Thank you ! Have a safe Journey";

mm.IsBodyHtml = false;

SmtpClient smtp = new SmtpClient();

smtp.Host = "smtp.gmail.com";

smtp.Port = 587;

smtp.EnableSsl = true;

NetworkCredential nc = new NetworkCredential("railwayreservationsystemmail@gmail.com", "chfxpbtcfjfobhlv");

smtp.UseDefaultCredentials = true;

smtp.Credentials = nc;

smtp.Send(mm);

//TempData["Message"] = "Ticket Booked !";

//redirect to payment gateway

return RedirectToAction("Success");

}

else

{

return RedirectToAction("Failed");

}

}

public ActionResult Success(OrderModel e)

{

return View();

}

public ActionResult Failed()

{

return View();

}

}

}

ResController.cs

﻿using RailwayReservationMVC.Models.DAL;

using RailwayReservationMVC.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Newtonsoft.Json.Linq;

using System.Xml.Linq;

namespace RailwayReservationMVC.Controllers

{

public class ResController : Controller

{

private IRailwayRepository<TrainDetails> TrainObj;

private IRailwayRepository<Reservation> ReservationObj;

public ResController()

{

this.TrainObj = new RailwayRepository<TrainDetails>();

this.ReservationObj = new RailwayRepository<Reservation>();

}

//reservation page

public ActionResult Index()

{

List<TrainDetails> trainDetails = TrainObj.GetModel().ToList();

ViewData["TrainDetailS"] = new SelectList(trainDetails, dataValueField: "SourceStation", dataTextField: "SourceStation");

ViewData["TrainDetailDesti"] = new SelectList(trainDetails, dataValueField: "DestinationStation", dataTextField: "DestinationStation");

return View();

}

[HttpPost]

public ActionResult Index(Reservation res)

{

List<TrainDetails> trainDetails = TrainObj.GetModel().ToList();

ViewData["TrainDetailS"] = new SelectList(trainDetails, dataValueField: "SourceStation", dataTextField: "SourceStation");

ViewData["TrainDetailDesti"] = new SelectList(trainDetails, dataValueField: "DestinationStation", dataTextField: "DestinationStation");

if (res != null)

{

Session["ResName"] = res.Res\_Name;

Session["ResGender"] = res.Res\_Gender;

Session["SourceStation"] = res.TrainDetails.SourceStation;

Session["DestinationStation"] = res.TrainDetails.DestinationStation;

Session["Quota"] = res.QuotaType;

Session["seatno"] = res.Seat\_No;

//res.User\_Id = (int)Session["user\_id"];

//var train\_Id = from id in trainDetails where id.SourceStation == res.TrainDetails.SourceStation select id;

Session["Date"] = res.Res\_Date;

//ReservationObj.InsertModel(res);

//ReservationObj.Save();

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

}

return View();

}

public ActionResult CancelPNR()

{

return View();

}

[HttpPost]

public ActionResult CancelPNR(int ModelId)

{

// long i = Convert.ToInt64(PNR);

ReservationObj.DeleteModel1(ModelId

);

ReservationObj.Save();

TempData["cancelmessage"] = "Your Ticket has been cancel !";

return View();

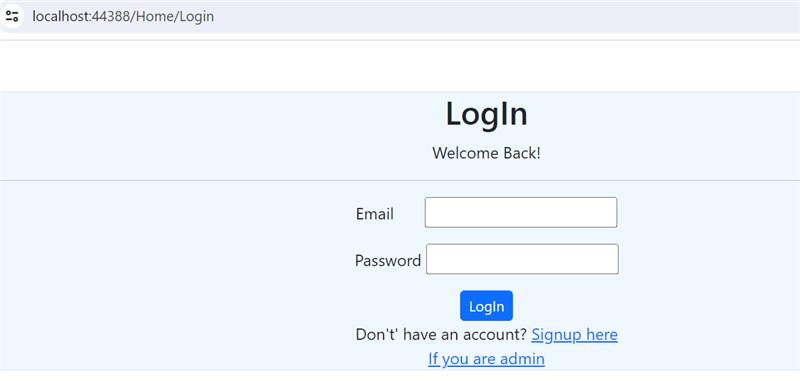
}

}

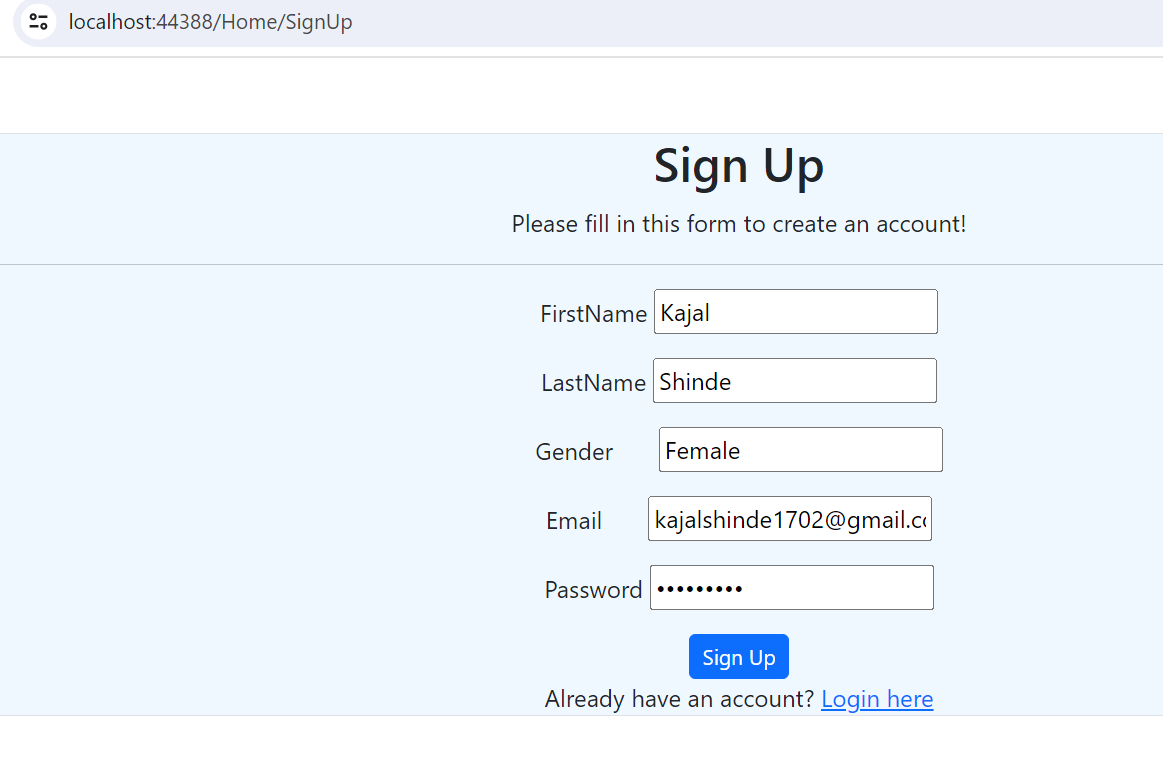
}

## OUTPUT:-

1.Login



2.SignUp



## FEASIBILITY STUDY:-

* **Technical Feasibility:-**

* The Hardware requirements to support the system are readily available.

1. System is developed by .Net technology
2. Backup and restore facility**.**

❖ **Operational Feasibility:-**

Railway Reservation System has various kinds of information that helps regarding booking of railway tickets.

This system contains the information about the passenger.

Users will be able to search the train, train fare, arrival and departure time of the train and they can also book the ticket by using online payment and after booking the ticket if the user wants to cancel the ticket, they can cancel that ticket.

## ❖ Economical Feasibility:-

❖ The system is created using existing hardware and software.

# LIMITATIONS

## LIMITATIONS:-

1. Limited capacity for simultaneous bookings.
2. Time constraints for reservations during peak hours.
3. Potential delays due to network connectivity issues.
4. Restrictions on the number of tickets per transaction.

Top of Form