|  |
| --- |
| Karnel Travel Guide |
| APTECH F.B AREA Center |
|  |
|  |

**E-Project**

**Karnel Travel Guide**

Submitted To:

eProjects@aglsm.com

Project Instructor:

Sir Faizan

Group Members:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Farhad Ahmed Tahir**  **Student1313074**  **Zohaib Khan**  **Student1313679**  **Farheen Safdar**  **Student1294557** | |  | | |  |  | | --- | --- | |  |  | |  | |  |  | | |  |
|  |
|  |



PROJECT

CERTIFICATION

**(Farhad 1313074 Hasnain 1314705 Farheen 1294557)**

## Has successfully Design and Developed

# EXCELLENT COACHING

## Submitted by:

# Farhad , Hasnain & Farheen

## Sir Faizan 5th/July/2022

***Acknowledgement:***



We like to thank many people who helped us in completing this project. Formost I would like to thank Almighty Allah for giving us strength and ability to complete this project. We would like to express our gratitude to all those who gave us the possibility to complete this website and we want to thank the Department of faculty for helping us to do the necessary research. We furthermore, thanks to our class teacher Sir Faizan our Great Teacher and also our Faculty Instractor

Sir Faizan for giving us completes support and advising us in the project.

Thank You.

**TABLE OF CONTENTS**

[CERTIFICATION OF COMpletion……………………………………………………………………….](#_Toc202078756)

Project plan……………………………………………………………………………..

[TABLE OF CONTENTS………………………………………………………………………………](#_Toc202078757)

[PROBLEM DEFINITION0…………………………………………………………………….…..](#_Toc202078758)

[Algorithms……………………………………………………………………………………](#_Toc202078760)

Task sheet…………………………………………………………………………………………..

[GUI Standards Documents……………………………………………………………………………](#_Toc202078765)

[interface design document………………………………………………………………………..](#_Toc202078766)

Project review and monitoring report…………………………………………………………………………………..

unit testing check list……………………………………………………………………………………………

Final check list……………………………………………………………………………….…

Source code……………………………………………………………………………….…

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Task | Actual Start Date | Actual Days | Team Mate Name | Status |
| 1. | Information Collecting | 30-May-2022 | 1 | Zohaib, Farheen | **Completed** |
| 3. | Making Layout/Pages | **28-May-2022** | 6 | Farhad | **Completed** |
| 4. | Connecting Links | 30-May-2022 | 2 | Farhad | **Completed** |
| 5. | Website Testing | 8-June-2022 | 3 | Farhad,Zohaib,Farheen | **Completed** |
| 6. | Images Collection | 1- June-2022 | 3 | Farhad,Zohaib,Farheen | **Completed** |
| 7. | Documentation | 15-June-2021 | 2 | Farhad | **Completed** |

**Task Sheet**

**LIST OF VALIDATION**

|  |  |
| --- | --- |
| **Option** | **Validation** |
| Fields In Form should not be Null | **Yes** |
| Homepage Has Forward Only Link | **Yes** |
| All Webpages have link back to Homepage | **Yes** |
| Request Quote Form Included | **Yes** |
| All Pages opens in mainframe (iframe) | **Yes** |
| All Functions Are Compatible with Google Chrome | **Yes** |

# SUBMISSION CHECKLIST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **LIST OF ITEMS** | **YES** | **NO** | **COMMENTS** |
| 1. | Do All Pages Linked together | **Yes** |  |  |
| 2. | All Hyperlinks working | **Yes** |  |  |
| 3. | Site Browser Compatible | **Yes** |  | **Tested Google Chrome,Firefox,IE8** |
| 4. | Project Zip File | **Yes** |  | **Created using WinRAR** |

**User Guide:**

## *Introduction : –*

“Man cannot discover new oceans unless he has the courage to lose sight of the shore.” – Andre Gide

Travling around the world is everyones dream but most people don`t know how to achive it or where to get tickets, visas, hotel reservations etc so to ease all these problems, we have proposed a solution as Karnel Travel Guide.

***Existing Scenario :*** *–*

* “Man cannot discover new oceans unless he has the courage to lose sight of the shore.” – Andre Gide
* Travling around the world is everyones dream but most people don`t know how to achive it or where to get tickets, visas, hotel reservations etc so to ease all these problems, we have proposed a solution as Karnel Travel Guide.
* *Proposed Solution :* –
* Karnel Travel Guide is a web portal where you can book your tours in advance, know your dates and timings.
* User can know everything about their tours rates and time just sitting on your couch.
* Just go to web portal for Karnel Travel Guide and register yourself.
* Karnel Travel Guide in MVC is a very user-friendly project and can be accessed from anywhere.
* There is no need to go to the Travel Agencies and wait for your turn to book tours.
* User can now just track everything about a tour just by clicking on to it.

. *Functional Requirements :* ***–***

**Karnel Travel Guide system features:**

* Login and Registration for users.
* Domestic Tours.
* Rates of each tour.
* Transportation of each city.
* Detailes of every tour/travel.
* Contact us with user friendly email client.
* Cart.
* Order.

**How do we start accessing the Sample eProject?**

The students at the center are expected to complete this eProject and send complete documentation with source code within the time allotted to eProjects Tea.

Looking forward to a positive response from your end!!

**Objectives of the project**

The Objective of this program is to give a sample project to work on real life projects. These applications help you build a larger more robust application.

The objective is not to teach you concepts but to provide you with a real life scenario and help you create applications using the tools.

You can revise them before you start with the project.

It is very essential that a student has a clear understanding of the subject.

Kindly get back to eProjects Team in case of any doubts regarding the application or its objectives.

.

**Problem Statement**

Problem statement is attached separately in a file “**Karnel Travel Guide**”

**Standards plan:**

Every code block must have comments.

The logic of the program needs to be explained. Proper documentation should be maintained.

Complete Project Report along with synopsis, code and documentation should be prepared.

**Documentation:**

No project is complete without documentation. In fact, it is one of the most important activities during the development of a project. The documentation of an ideal project will be in the form of a project report comprising of the following documents:

* Certification of completion.
* Table of Contents.
* Problem Definition.
* Customer Requirement Specification.
* Project Plan.
* Algorithms.
* E-R Diagrams.
* GUI Standards Document.
* Interface Design Document.
* Task Sheet
* Project Review and Monitoring Report.
* Unit Testing Check List.
* Final Check List.

Hardware/ Software Requirements

**Hardware**

* A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better
* 128 Megabytes of RAM or better

**Operating System**

* LINUX / Windows 2000 Server (or higher if possible)

**Software**

* ASP.NET
* MVC
* My SQL Server
* Bootstrap

using Karnel\_Travel\_Guide.Models;

using Karnel\_Travel\_Guide.ViewModel;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.SqlClient;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Karnel\_Travel\_Guide.Controllers

{

public class AdminController : Controller

{

Karnel\_Travel\_GuideEntities db = new Karnel\_Travel\_GuideEntities();

// GET: Admin

public ActionResult Index()

{

return View();

}

public ActionResult Dashboard()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

var tc = db.tb\_transportation.Count() + db.tb\_tour.Count() + db.tb\_restaurant.Count() + db.tb\_resorts.Count() + db.tb\_packages.Count() + db.tb\_hotels.Count() ;

TempData["Product\_Count"] = tc;

TempData["User\_Count"] = db.tb\_user.Count();

TempData["Order\_Count"] = db.tb\_orders.Count();

TempData["Order\_Total"] = db.tb\_orders.Sum(p => p.o\_total);

return View();

}

}

public ActionResult Admin\_Login()

{

return View();

}

[HttpPost]

public ActionResult Admin\_Login(tb\_admin a)

{

var data = db.tb\_admin.Where(model => model.ad\_email == a.ad\_email && model.ad\_password == a.ad\_password).FirstOrDefault();

if (data != null)

{

Session["Admin\_id"] = data.ad\_id.ToString();

Session["Admin\_name"] = data.ad\_name.ToString();

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

}

else

{

ModelState.Clear();

return View();

}

}

public ActionResult All\_Category()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

var data = db.tb\_category.ToList();

return PartialView(data);

}

}

public ActionResult Category\_delete(int id)

{

var data = db.tb\_category.Where(m => m.c\_id == id).SingleOrDefault();

db.Entry(data).State = EntityState.Deleted;

db.SaveChanges();

return RedirectToAction("All\_Category", "Admin");

}

public ActionResult Category()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

return View();

}

}

[HttpPost]

public int Cat(string cat\_name)

{

if (cat\_name != null)

{

tb\_category c = new tb\_category();

c.c\_name = cat\_name;

db.tb\_category.Add(c);

db.SaveChanges();

return 1;

}

else

{

return 0;

}

}

public ActionResult All\_Product()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

var tables = new ProductViewModel

{

Tb\_Transportations = db.tb\_transportation.ToList(),

Tb\_Tours = db.tb\_tour.ToList(),

Tb\_Restaurants = db.tb\_restaurant.ToList(),

Tb\_Resorts = db.tb\_resorts.ToList(),

Tb\_Packages = db.tb\_packages.ToList(),

Tb\_Hotels = db.tb\_hotels.ToList()

};

//return View(tables);

return PartialView(tables);

}

}

[HttpPost]

public ActionResult Product\_delete(int id)

{

var data = db.tb\_transportation.Where(m => m.tr\_id == id).SingleOrDefault();

var data1 = db.tb\_tour.Where(m => m.to\_id == id).SingleOrDefault();

var data2 = db.tb\_restaurant.Where(m => m.rest\_id == id).SingleOrDefault();

var data3 = db.tb\_resorts.Where(m => m.reso\_id == id).SingleOrDefault();

var data4 = db.tb\_packages.Where(m => m.pac\_id == id).SingleOrDefault();

var data5 = db.tb\_hotels.Where(m => m.h\_id == id).SingleOrDefault();

string image = Request.MapPath(data.tr\_images.ToString());

string image1 = Request.MapPath(data1.to\_images.ToString());

string image2 = Request.MapPath(data2.rest\_images.ToString());

string image3 = Request.MapPath(data3.reso\_images.ToString());

string image4 = Request.MapPath(data4.pac\_images.ToString());

string image5 = Request.MapPath(data5.h\_images.ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

if (System.IO.File.Exists(image1))

{

System.IO.File.Delete(image1);

}

if (System.IO.File.Exists(image2))

{

System.IO.File.Delete(image2);

}

if (System.IO.File.Exists(image3))

{

System.IO.File.Delete(image3);

}

if (System.IO.File.Exists(image4))

{

System.IO.File.Delete(image4);

}

if (System.IO.File.Exists(image5))

{

System.IO.File.Delete(image5);

}

db.Entry(data).State = EntityState.Deleted;

db.Entry(data1).State = EntityState.Deleted;

db.Entry(data2).State = EntityState.Deleted;

db.Entry(data3).State = EntityState.Deleted;

db.Entry(data4).State = EntityState.Deleted;

db.Entry(data5).State = EntityState.Deleted;

db.SaveChanges();

return RedirectToAction("All\_Product", "Admin");

}

public ActionResult Product()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

return PartialView();

}

}

public PartialViewResult Transportation()

{

return PartialView("\_Producttr");

}

public PartialViewResult Tour()

{

return PartialView("\_Productto");

}

public PartialViewResult Restaurant()

{

return PartialView("\_Productrest");

}

public PartialViewResult Packages()

{

return PartialView("\_Productpac");

}

public PartialViewResult Resorts()

{

return PartialView("\_Productreso");

}

public PartialViewResult Hotels()

{

return PartialView("\_Producth");

}

public PartialViewResult Accommodation()

{

return PartialView("\_Productac");

}

public ActionResult Logout()

{

Session.Abandon();

Session.Clear();

return RedirectToAction("Admin\_login", "Admin");

}

public ActionResult All\_Customers()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

var data = db.tb\_user.ToList();

return PartialView(data);

}

}

[HttpPost]

public ActionResult Customer\_delete(int id)

{

var data = db.tb\_user.Where(m => m.u\_tb == id).SingleOrDefault();

db.Entry(data).State = EntityState.Deleted;

db.SaveChanges();

return RedirectToAction("All\_Customers", "Admin");

}

public ActionResult Order()

{

if (Session["Admin\_id"] == null)

{

return RedirectToAction("Admin\_login", "Admin");

}

else

{

var data = db.tb\_orders.OrderByDescending(x => x.o\_id).ToList();

return PartialView(data);

}

}

[HttpPost]

public ActionResult Order\_Update(string status, int? id)

{

if (status != null && id != null)

{

var data = db.tb\_orders.Where(x => x.o\_id == id).SingleOrDefault();

data.o\_id = Convert.ToInt32(id);

data.o\_billing\_city = data.o\_billing\_city;

data.o\_billing\_name = data.o\_billing\_name;

data.o\_date = data.o\_date;

data.o\_items = data.o\_items;

data.o\_payment = data.o\_payment;

data.o\_status = status;

data.o\_total = data.o\_total;

data.o\_quantity = data.o\_quantity;

data.o\_foreign\_key = data.o\_foreign\_key;

db.Entry(data).State = EntityState.Modified;

db.SaveChanges();

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

}

else

{

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

}

}

public ActionResult Producttr()

{

return PartialView();

}

[HttpPost]

public ActionResult Producttr(tb\_transportation p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.tr\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

p.tr\_discount = "no";

db.tb\_transportation.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Productto()

{

return PartialView();

}

[HttpPost]

public ActionResult Productto(tb\_tour p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.to\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

p.to\_discount = "no";

db.tb\_tour.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Productrest()

{

return PartialView();

}

[HttpPost]

public ActionResult Productrest(tb\_restaurant p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.rest\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

db.tb\_restaurant.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Productreso()

{

return PartialView();

}

[HttpPost]

public ActionResult Productreso(tb\_resorts p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.reso\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

db.tb\_resorts.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Productpac()

{

return PartialView();

}

[HttpPost]

public ActionResult Productpac(tb\_packages p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.pac\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

p.pac\_discount = "no";

db.tb\_packages.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Producth()

{

return PartialView();

}

[HttpPost]

public ActionResult Producth(tb\_hotels p)

{

string filename = Path.GetFileNameWithoutExtension(p.ImageFile.FileName);

string extension = Path.GetExtension(p.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

p.h\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

p.ImageFile.SaveAs(filename);

db.tb\_hotels.Add(p);

db.SaveChanges();

ModelState.Clear();

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

}

else

{

ModelState.Clear();

//return PartialView();

return Content("<script language='javascript' type='text/javascript'>alert('Error');</script>");

}

}

public ActionResult Edittr(int id)

{

var data = db.tb\_transportation.Where(m => m.tr\_id == id).FirstOrDefault();

TempData["Images"] = data.tr\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Edittr(tb\_transportation v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.tr\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.tr\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

public ActionResult Editto(int id)

{

var data = db.tb\_tour.Where(m => m.to\_id == id).FirstOrDefault();

TempData["Images"] = data.to\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Editto(tb\_tour v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.to\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.to\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

public ActionResult Editrest(int id)

{

var data = db.tb\_restaurant.Where(m => m.rest\_id == id).FirstOrDefault();

TempData["Images"] = data.rest\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Editrest(tb\_restaurant v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.rest\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.rest\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

public ActionResult Editreso(int id)

{

var data = db.tb\_resorts.Where(m => m.reso\_id == id).FirstOrDefault();

TempData["Images"] = data.reso\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Editreso(tb\_resorts v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.reso\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.reso\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

public ActionResult Editpac(int id)

{

var data = db.tb\_packages.Where(m => m.pac\_id == id).FirstOrDefault();

TempData["Images"] = data.pac\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Editpac(tb\_packages v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.pac\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.pac\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

public ActionResult Edith(int id)

{

var data = db.tb\_hotels.Where(m => m.h\_id == id).FirstOrDefault();

TempData["Images"] = data.h\_images.ToString();

TempData.Keep();

return PartialView(data);

}

[HttpPost]

public ActionResult Edith(tb\_hotels v)

{

if (v.ImageFile != null)

{

string filename = Path.GetFileNameWithoutExtension(v.ImageFile.FileName);

string extension = Path.GetExtension(v.ImageFile.FileName);

if (extension == ".jpg" || extension == ".png" || extension == ".jpeg")

{

filename = filename + extension;

v.h\_images = "~/Images/" + filename;

filename = Path.Combine(Server.MapPath("~/Images/"), filename);

v.ImageFile.SaveAs(filename);

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

string image = Request.MapPath(TempData["Images"].ToString());

if (System.IO.File.Exists(image))

{

System.IO.File.Delete(image);

}

ModelState.Clear();

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

}

else

{

ModelState.Clear();

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

}

}

else

{

v.h\_images = TempData["Images"].ToString();

db.Entry(v).State = EntityState.Modified;

db.SaveChanges();

ModelState.Clear();

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

}

}

}

}







