PROJECT REPORT

Tripppy

Trip Booking Database Management System

26.04.2021

Team Members:

RA1811003010189- Lennox Sherwin Sweeton

RA1811003010190- Aayush Dua

RA1811003010197- Sandesh Jain

RA1811003010203- Naman Sharma

RA1811003010205- A Kavin Chandar

Index

Introduction

What is Tripppy?

+ Tripppy is a Trip Booking Application and as such functions as an online booking system

+ An online booking system feeds a huge amount of guest data right into your system: how many guests are travelling, when they come, what amenities they prefer, where they live, and so forth. That's very useful for you to know, and will help you improve both customer service and marketing. Therefore, we have designed said system to do the following and more...

The Trip Booking Management System is a web based application. The main purpose of “Tripppy” is to provide a convenient way for a customer to book hotels, flight, train and bus for the tour purposes. The objective of this project is to develop a system that automates the processes and activities of a travel agency . In this project, we will make an easier task of searching places and for booking trains, flights or buses. In the present system a customer has to approach various agencies to find details of places and to book tickets. This often requires a lot of time and effort. We provide approach skills to critically examine how a tourist visits and its ability to operate in an appropriate way when dealing with the consequences of tourism, locally , regionally and nationally including visitor security and ecological influences. It is tedious for a customer to plan a particular journey and have it executed properly. The project “Tours and Travels Management System” is developed to replace the currently existing system, which helps in keeping record of customer details of destination as well as payment received. Tourism marketers are faced with a complex environment resulting from unprecedented growth in the tourism industry over the last fifty years. As the phenomenon of tourism has grown, so have the interests of destinations in attracting their share of visitors. Destination choices available to consumers have proliferated. Today’s tourism marketers must influence consumer decision making in an increasingly complex and competitive global marketplace. Tourism is a complicated setting involving a diverse group of active stakeholders who each have different interests in the tourism market. For destination stakeholders such as Destination Marketing Organisations (DMOs), accommodation providers, and activity operators to survive in an increasingly competitive environment, it is essential that a consistent approach is used by all tourism stakeholders operating within a single destinationThe importance of segmentation in tourism is widely acknowledged .To date research has assisted us to understand which bases can be used by tourism destinations to effectively for a destination using visitor data. Little research attention has been directed towards understanding how the tourism stakeholders segment their markets. As a result we do not know how tourism stakeholders segment a market for managerial and marketing purposes and whether tourism stakeholder segments mirror the segments defined by DMOs. This case study will contribute to the literature by presenting tourism stakeholder views for one tourism destination. It will identify similarities or discrepancies between the segments defined by the DMO and the segments used by tourism stakeholders for managerial and marketing purposes.

System Requirements

Hardware Requirements:

Software Requirements:

OPERATING SYSTEM :Any Web Browser PHP ver. +5.4.4 WEB SERVER :Apache with PHP and MySQL

User Characteristics :

a)Every user.

b)Should be comfortable with basics working on the computer

c)Must carry a login ID and Password used for authentication.

Login id

Admin I'd- admin@gmail.com

Password- admin

User Id and Password- [For user you can Signup and get your own login id & pass]

Tools Used : MAMP

System Design and Analysis

System design is essential to develop a model of a system before writing any software that is used to control the system or to interact with it during the design process. We try to develop system models at different levels of abstraction. Design process invoices data flow diagram and data structure including library functioned in the program. The project developed using the below objects:

Project Description

This application in develop to provide best travelling services to the customers and travel agents we have developed tours and travel management item to provide a search platform where tourist can find their tour places according to their choices This system also help to promote responsible and interesting tourism so that people can enjoy their holidays at their favorable places system also help to develop tourism with different cultures so that they enrich the tourism experience and build pride We develop this system to create and promote forms of tourism provide healthy interaction opportunities for tourists and locals and increase better understanding different cultures, customs, lifestyles, traditional knowledge and believes. This system also provide a better way to connect with various events This system also gives tours related information like which places de tourist attractions, provinces Tourist can also get the Map and navigation system and temperature and weather information Tourist can also book tours through our tours and travels management system system also keeps a history of visited places of its users.

System Development Phase

Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design and implementation, and are explained in the section below. There are several Systems Development Life Cycle Models in existence. The oldest model that was originally regarded as the Systems Development Life Cycle is the waterfall model sequence of stages in which the output of each stage becomes the input for the next These stages generally follow the same basic step but many different waterfall methodologies give the step different names and the number of steps seem to vary between 4 and 7 There is no definitively correct Systems Development Life Cycle model in the stars can be characterized and divided several steps . Initiation Phase The Initiation Phase begins when a business or identifies a need or an opportunity. The purpose of the Initiation Phase is to Identity and validate an opportunity to improve business accomplishments of the an n o deficiency related to a business need identify significant assumptions and constraints on solutions t that need Recommend the explosion of alternative concepts and methods to satisfy the need including questioning the need for technology , will change in the business process offer a solution Assure executive business and executive technical sponsorship

System Concept Development Phase

The System Concept Development Phase begins after business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO. The purpose of the System Concept Development Phase is to: Determine the feasibility and appropriateness of the alternatives Identity system interfaces Identify basic functional and data requirements to satisfy the business need Establish system boundaries identity goals, objectives, critical success factors, and performance measures Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements Assess project risks Identify and initiate risk mitigation actions, and Develop high-level technical architecture process models, data models and a concept of operations

Tech Stack

Adobe XD

Adobe XD was used to design the UI/UX of the product with an early prototype to give form to the idea.

HTML / CSS / Javascript

Html , CSS and Javasript were used to create the Front-end of the application

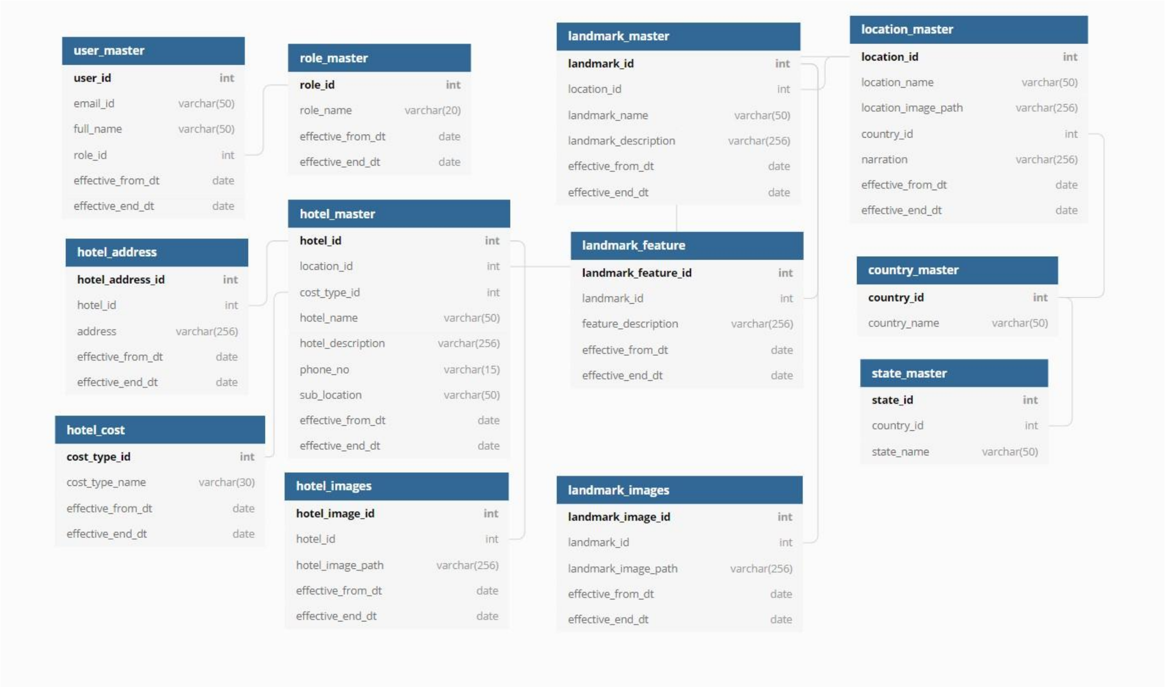
PHP

PHP is a widely used general purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose. PHP code is embedded into the HTML source document and interpreted by the web server with the PHP processor module which generates web page documents. PHP source code compiled on the fly to an internal from that can be executed by the PHP engine in order to speed up execution time and not have to compile the PHP source code every time the webpage is access , PHP scripts can also deployed in executable format using a PHP compiler. PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic websites that interact with the user all mixed information PHP offers many advantages. It is fast, stable, secure, is easy to use and open source (free). PHP code inserted directly into the HTML that makes up a website When a visitor comes to the website the code is executed Because a PHP is a server side technology, the user does not bed any special browser or plans to see the PHP in action. Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of lines such as graphics, XML ,encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them in the executable or they can create their own executable and load it using PHPs dynamic loading mechanism. A huge advantage that PHP offers is its community. Since PHP is a huge advantage that PHP offers is its community. Since PHP is an open source project, the PHP community is willing to share. If you're looking for a particular script chances are another user has already created something similar Check within the PHP community for availability Likewise, if you have created a function that others might enjoy, be sure to post the code for others. PHP scripting block always starts with A PHP scripting block can be placed anywhere in the document On servers with shorthand support enabled you can start a scripting block with . For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

MY SQL

MySQL is an open-source relational database management system(RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter and "SQL, the abbreviation for Structured Query Language The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB now owned by Oracle Corporation For proprietary use, several paid editions are available, and offer additional functionality. MySQL was created by a Swedish .company, MySQL added by David Axmark, Allan Larsson and Michael ``Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994 The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from SQL based on the low-level language SAM, which the creators considered too slow and inflexible They created a new SQL interface, while keeping the same API as MYSQL MySQL written in C and C ++ .Its SQL parser is written in yacc, but it uses a homebrewed lexical analyzer. MySQL dump is a logical backup tool included with both community and enterprise editions of MYSQL. It supports backing up from all storage engines. MySQL Enterprise Backup is a hot backup utility included as part of the MySQL Enterprise subscription from Oracle, offering native InnoDB hot backup, as well as backup for other storage engines Extra Backup is an open-source MYSQL hot backup software program Features include hot, non locking backups for In no DB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of UO operations per second, etc. MySQL Fabric is an integrated system for managing a collection of MySQL servers, and a framework on top of which high availability and database sharing is built MySQL Fabric is open source, and supports procedure execution in the presence of failure, providing an execution model usually called resilient execution MySQL client libraries are extended so they are hiding the complexities or handling fail over in the event of a server failure, as well as correctly dispatching transactions to the shards. PhpMyAdmin is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser It can perform various tasks such as creating modifying or deleting databases, tables, fields or rows, executing SQL statements, or managing users and permissions. The software, which is available in 78 languages, is maintained by ThephpMyadminProj

Class Diagram



user_master	
user_id	int
email_id	varchar(50)
full_name	varchar(50)
role_id	int
effective_from_dt	date
effective_end_dt	date

role_master	
role_id	int
role_name	varchar(20)
effective_from_dt	date
effective_end_dt	date

landmark_master	
landmark_id	int
location_id	int
landmark_name	varchar(50)
landmark_description	varchar(256)
effective_from_dt	date
effective_end_dt	date

location_master	
location_id	int
location_name	varchar(50)
location_image_path	varchar(256)
country_id	int
narration	varchar(256)
effective_from_dt	date
effective_end_dt	date

hotel_address	
hotel_address_id	int
hotel_id	int
address	varchar(256)
effective_from_dt	date
effective_end_dt	date

hotel_master	
hotel_id	int
location_id	int
cost_type_id	int
hotel_name	varchar(50)
hotel_description	varchar(256)
phone_no	varchar(15)
sub_location	varchar(50)
effective_from_dt	date
effective_end_dt	date

landmark_feature	
landmark_feature_id	int
landmark_id	int
feature_description	varchar(256)
effective_from_dt	date
effective_end_dt	date

country_master	
country_id	int
country_name	varchar(50)

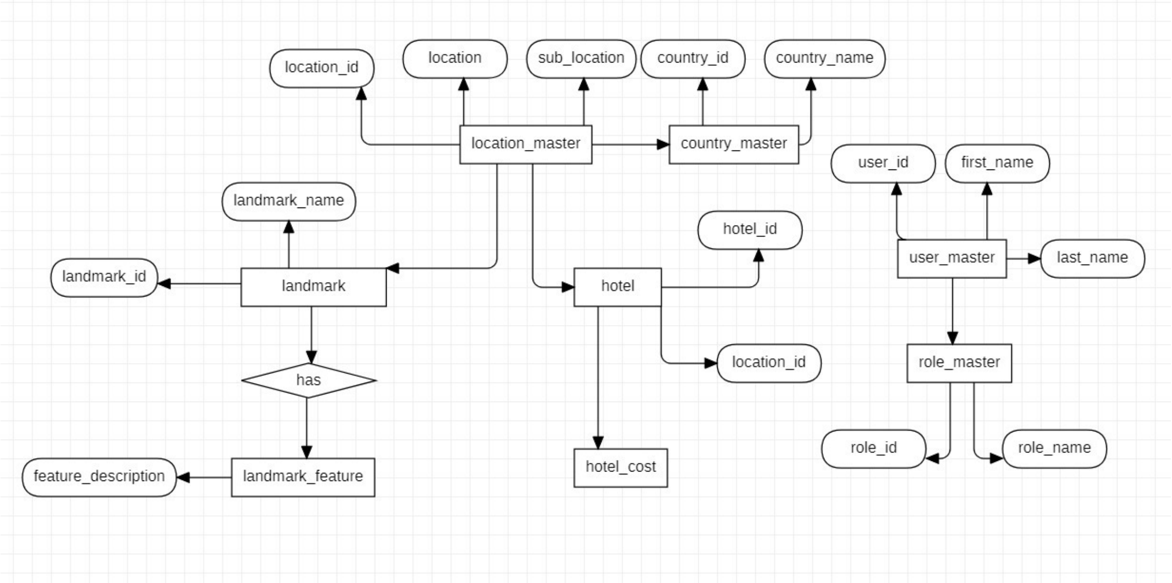
state_master	
state_id	int
country_id	int
state_name	varchar(50)

hotel_cost	
cost_type_id	int
cost_type_name	varchar(30)
effective_from_dt	date
effective_end_dt	date

hotel_images	
hotel_image_id	int
hotel_id	int
hotel_image_path	varchar(256)
effective_from_dt	date
effective_end_dt	date

landmark_images	
landmark_image_id	int
landmark_id	int
landmark_image_path	varchar(256)
effective_from_dt	date
effective_end_dt	date

ER Model



Implementation Phase

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto the production computers, and integration of the system onto daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

SQL Commands

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
-- Host: localhost:3306
-- Generation Time: Apr 22, 2021 at 12:23 PM
-- Server version: 5.7.24
-- PHP Version: 7.2.14

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

 Database: `trip_booking`
-- Table structure for table `country_master`
```

```
CREATE TABLE `country_master` (
  `country_id` int(11) NOT NULL,
  `country_name` varchar(50) DEFAULT NULL,
  `country_iso_code` varchar(5) DEFAULT NULL,
  `country_isd_code` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `country_master`

INSERT INTO `country_master` (`country_id`, `country_name`, `country_iso_code`, `country_isd_code`)
VALUES
(1, 'Afghanistan', 'AF', '93'),
(2, 'Albania', 'AL', '355'),
(3, 'Algeria', 'DZ', '213'),
(4, 'American Samoa', 'AS', '1-684'),
(5, 'Andorra', 'AD', '376'),
(6, 'Angola', 'AO', '244'),
(7, 'Anguilla', 'AI', '1-264'),
(8, 'Antarctica', 'AQ', '672'),
(9, 'Antigua and Barbuda', 'AG', '1-268'),
(10, 'Argentina', 'AR', '54'),
(11, 'Armenia', 'AM', '374'),

-- Table structure for table `hotel_address`

CREATE TABLE `hotel_address` (
  `hotel_address_id` int(11) NOT NULL,
  `hotel_id` int(11) DEFAULT NULL,
  `address` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Table structure for table `hotel_cost`

CREATE TABLE `hotel_cost` (
  `cost_type_id` int(11) NOT NULL,
  `cost_type_name` varchar(30) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Table structure for table `hotel_images`

CREATE TABLE `hotel_images` (
  `hotel_image_id` int(11) NOT NULL,
  `hotel_id` int(11) DEFAULT NULL,
  `hotel_image_path` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Table structure for table `hotel_master`

CREATE TABLE `hotel_master` (
  `hotel_id` int(11) NOT NULL,
  `location_id` int(11) DEFAULT NULL,
  `cost_type_id` int(11) DEFAULT NULL,
  `hotel_name` varchar(50) DEFAULT NULL,
  `hotel_description` varchar(150) DEFAULT NULL,
  `hotel_rating` int(11) NOT NULL,
  `phone_no` varchar(15) DEFAULT NULL,
  `hotel_image_path` varchar(100) NOT NULL,
  `hotel_cost` decimal(10,2) NOT NULL,
  `sub_location` varchar(50) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Table structure for table `landmark_feature`

CREATE TABLE `landmark_feature` (
  `landmark_feature_id` int(11) NOT NULL,
  `landmark_id` int(11) DEFAULT NULL,
  `feature_description` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Table structure for table `landmark_images`

CREATE TABLE `landmark_images` (
  `landmark_image_id` int(11) NOT NULL,
  `landmark_id` int(11) DEFAULT NULL,
  `landmark_image_path` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Table structure for table `landmark_master`

CREATE TABLE `landmark_master` (
  `landmark_id` int(11) NOT NULL,
  `location_id` int(11) DEFAULT NULL,
  `landmark_name` varchar(50) DEFAULT NULL,
  `landmark_description` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Table structure for table `location_master`

CREATE TABLE `location_master` (
  `location_id` int(11) NOT NULL,
  `location_name` varchar(50) DEFAULT NULL,
  `location_image_path` varchar(256) DEFAULT NULL,
  `country_id` int(11) DEFAULT NULL,
  `narration` varchar(256) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `location_master`

INSERT INTO `location_master` (`location_id`, `location_name`, `location_image_path`, `country_id`,
`narration`, `effective_from_dt`, `effective_end_dt`) VALUES
(1, 'Goa', 'images/bg.jpg', 99, 'Goa is a good place', '2121-04-14', NULL),
(2, 'Kodaikanal', 'images/bg2.jpg', 99, 'jhskdd', '2121-04-14', '2121-04-14'),
(3, 'Ooty', 'images/bg3.jpg', 1, 'Hello', '2121-04-14', NULL),
(4, 'Udaipur', 'images/AIsol.png', 1, 'joking', '2121-04-14', NULL),
(5, 'Spiti', 'images/AIsol.png', 99, 'testing', '2121-04-20', NULL),
(6, 'Alleppey', '/tripppy/images/AIsol.png', 1, '', '2121-04-20', NULL),
(7, 'Coorg', '/tripppy/images/AIsol.png', 1, 'dskjlk', '2121-04-22', NULL),
(8, 'Katch', 'images/bg.jpg', 1, 'hghgh', '2121-04-22', NULL);




-- Table structure for table `role_master`
```

```
CREATE TABLE `role_master` (
  `role_id` int(11) NOT NULL,
  `role_name` varchar(20) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `role_master`

INSERT INTO `role_master` (`role_id`, `role_name`, `effective_from_dt`, `effective_end_dt`) VALUES
(1, 'Admin', '2021-03-26', NULL),
(2, 'User', '2021-03-26', NULL);



-- Table structure for table `state_master`

CREATE TABLE `state_master` (
  `state_id` int(11) NOT NULL,
  `country_id` int(11) DEFAULT NULL,
  `state_name` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `state_master`

INSERT INTO `state_master` (`state_id`, `country_id`, `state_name`) VALUES
(1, 99, 'ANDHRA PRADESH'),
(2, 99, 'ASSAM'),
(3, 99, 'ARUNACHAL PRADESH'),
(4, 99, 'GUJRAT'),
(5, 99, 'BIHAR'),
(6, 99, 'HARYANA'),
(7, 99, 'HIMACHAL PRADESH'),
(8, 99, 'JAMMU & KASHMIR'),
(9, 99, 'KARNATAKA'),
(10, 99, 'KERALA'),
(11, 99, 'MADHYA PRADESH'),
(12, 99, 'MAHARASHTRA'),
(13, 99, 'MANIPUR'),
(14, 99, 'MEGHALAYA'),
(15, 99, 'MIZORAM'),
(16, 99, 'NAGALAND'),
(17, 99, 'ORISSA'),
(18, 99, 'PUNJAB'),
(19, 99, 'RAJASTHAN'),
(20, 99, 'SIKKIM'),
(21, 99, 'TAMIL NADU'),
(22, 99, 'TRIPURA'),
(23, 99, 'UTTAR PRADESH'),
(24, 99, 'WEST BENGAL'),
(25, 99, 'GOA'),
(26, 99, 'PONDICHERY'),
(27, 99, 'LAKSHDWEEP'),
(28, 99, 'DAMAN & DIU'),
(29, 99, 'DADRA & NAGAR'),
(30, 99, 'CHANDIGARH'),
(31, 99, 'ANDAMAN & NICOBAR'),
(32, 99, 'UTTARANCHAL'),
(33, 99, 'JHARKHAND'),
(34, 99, 'CHATTISGARH'),
(35, 99, 'ASSAM');

-- Table structure for table `user_master`

CREATE TABLE `user_master` (
  `user_id` int(11) NOT NULL,
  `email_id` varchar(50) DEFAULT NULL,
  `password` varchar(256) DEFAULT NULL,
  `full_name` varchar(50) DEFAULT NULL,
  `role_id` int(11) DEFAULT NULL,
  `effective_from_dt` date DEFAULT NULL,
  `effective_end_dt` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `user_master`

INSERT INTO `user_master` (`user_id`, `email_id`, `password`, `full_name`, `role_id`,
`effective_from_dt`, `effective_end_dt`) VALUES
(1, 'admin@gmail.com', '21232f297a57a5a743894a0e4a801fc3', 'Admin Account', 1, NULL, NULL),
(3, 'aayush@gmail.com', '6bc80b9416b95aac0cf7757fc1bb1e65', 'Aayush Dua', 2, '2121-03-26', NULL);

-- Indexes for table `country_master`

ALTER TABLE `country_master`
  ADD PRIMARY KEY (`country_id`);

-- Indexes for table `hotel_address`

ALTER TABLE `hotel_address`
  ADD PRIMARY KEY (`hotel_address_id`),
  ADD KEY `hotel_id` (`hotel_id`);

-- Indexes for table `hotel_cost`
ALTER TABLE `hotel_cost`
  ADD PRIMARY KEY (`cost_type_id`);

-- Indexes for table `hotel_images`

ALTER TABLE `hotel_images`
  ADD PRIMARY KEY (`hotel_image_id`),
  ADD KEY `hotel_id` (`hotel_id`);

-- Indexes for table `hotel_master`

ALTER TABLE `hotel_master`
  ADD PRIMARY KEY (`hotel_id`),
  ADD KEY `location_id` (`location_id`),
  ADD KEY `cost_type_id` (`cost_type_id`);

-- Indexes for table `landmark_feature`

ALTER TABLE `landmark_feature`
  ADD PRIMARY KEY (`landmark_feature_id`),
  ADD KEY `landmark_id` (`landmark_id`);

-- Indexes for table `landmark_images`

ALTER TABLE `landmark_images`
  ADD PRIMARY KEY (`landmark_image_id`),
  ADD KEY `landmark_id` (`landmark_id`);

-- Indexes for table `landmark_master`

ALTER TABLE `landmark_master`
  ADD PRIMARY KEY (`landmark_id`),
  ADD KEY `location_id` (`location_id`);
-- Indexes for table `location_master`

ALTER TABLE `location_master`
  ADD PRIMARY KEY (`location_id`),
  ADD KEY `country_id` (`country_id`);

-- Indexes for table `role_master`
```

```
ALTER TABLE `role_master`
  ADD PRIMARY KEY (`role_id`);

-- Indexes for table `state_master`

ALTER TABLE `state_master`
  ADD PRIMARY KEY (`state_id`),
  ADD KEY `country_id` (`country_id`);

-- Indexes for table `user_master`

ALTER TABLE `user_master`
  ADD PRIMARY KEY (`user_id`),
  ADD KEY `role_id` (`role_id`);

-- AUTO_INCREMENT for table `country_master`

ALTER TABLE `country_master`
  MODIFY `country_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=250;

-- AUTO_INCREMENT for table `hotel_address`

ALTER TABLE `hotel_address`
  MODIFY `hotel_address_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=3;

-- AUTO_INCREMENT for table `hotel_cost`
ALTER TABLE `hotel_cost`
  MODIFY `cost_type_id` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT for table `hotel_images`
ALTER TABLE `hotel_images`
  MODIFY `hotel_image_id` int(11) NOT NULL AUTO_INCREMENT;
```

```
-- AUTO_INCREMENT for table `hotel_master`
ALTER TABLE `hotel_master`
  MODIFY `hotel_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=3;

-- AUTO_INCREMENT for table `landmark_feature`
ALTER TABLE `landmark_feature`
  MODIFY `landmark_feature_id` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT for table `landmark_images`
ALTER TABLE `landmark_images`
  MODIFY `landmark_image_id` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT for table `landmark_master`
ALTER TABLE `landmark_master`
  MODIFY `landmark_id` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT for table `location_master`
ALTER TABLE `location_master`
  MODIFY `location_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=9;

-- AUTO_INCREMENT for table `role_master`
ALTER TABLE `role_master`
  MODIFY `role_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=3;

-- AUTO_INCREMENT for table `state_master`
ALTER TABLE `state_master`
  MODIFY `state_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=36;

-- AUTO_INCREMENT for table `user_master`
ALTER TABLE `user_master`
  MODIFY `user_id` int(11) NOT NULL AUTO_INCREMENT,  AUTO_INCREMENT=4;
```

```
-- Constraints for table `hotel_address`

ALTER TABLE `hotel_address`
  ADD CONSTRAINT `hotel_address_ibfk_1` FOREIGN KEY (`hotel_id`) REFERENCES
`hotel_master` (`hotel_id`);

-- Constraints for table `hotel_images`

ALTER TABLE `hotel_images`
  ADD CONSTRAINT `hotel_images_ibfk_1` FOREIGN KEY (`hotel_id`) REFERENCES
`hotel_master` (`hotel_id`);

-- Constraints for table `hotel_master`

ALTER TABLE `hotel_master`
  ADD CONSTRAINT `hotel_master_ibfk_1` FOREIGN KEY (`location_id`) REFERENCES
`location_master` (`location_id`),
  ADD CONSTRAINT `hotel_master_ibfk_2` FOREIGN KEY (`cost_type_id`) REFERENCES
`hotel_cost` (`cost_type_id`);

-- Constraints for table `landmark_feature`

ALTER TABLE `landmark_feature`
  ADD CONSTRAINT `landmark_feature_ibfk_1` FOREIGN KEY (`landmark_id`) REFERENCES
`landmark_master` (`landmark_id`);

-- Constraints for table `landmark_images`

ALTER TABLE `landmark_images`
  ADD CONSTRAINT `landmark_images_ibfk_1` FOREIGN KEY (`landmark_id`) REFERENCES
`landmark_master` (`landmark_id`);

-- Constraints for table `landmark_master`

ALTER TABLE `landmark_master`
```

```
  ADD CONSTRAINT `landmark_master_ibfk_1` FOREIGN KEY (`location_id`) REFERENCES
`location_master` (`location_id`);

-- Constraints for table `location_master`

ALTER TABLE `location_master`
  ADD CONSTRAINT `location_master_ibfk_1` FOREIGN KEY (`country_id`) REFERENCES
`country_master` (`country_id`);

-- Constraints for table `state_master`

ALTER TABLE `state_master`
  ADD CONSTRAINT `state_master_ibfk_1` FOREIGN KEY (`country_id`) REFERENCES
`country_master` (`country_id`);

-- Constraints for table `user_master`

ALTER TABLE `user_master`
  ADD CONSTRAINT `user_master_ibfk_1` FOREIGN KEY (`role_id`) REFERENCES `role_master`
(`role_id`);
COMMIT;
```

User and Admin Privileges

Tripppy allows both user and admin privileges based on Log-In Information.

User End

User can view destinations and respective activities

Uses Data manipulation language to retrieve data from the database and display it to the front-end

The SELECT statements are used to display all the location infos as input fields.

Admin End

The admin page uses all the SQL statements required for a CRUD application(Create,Read,Update and Delete).

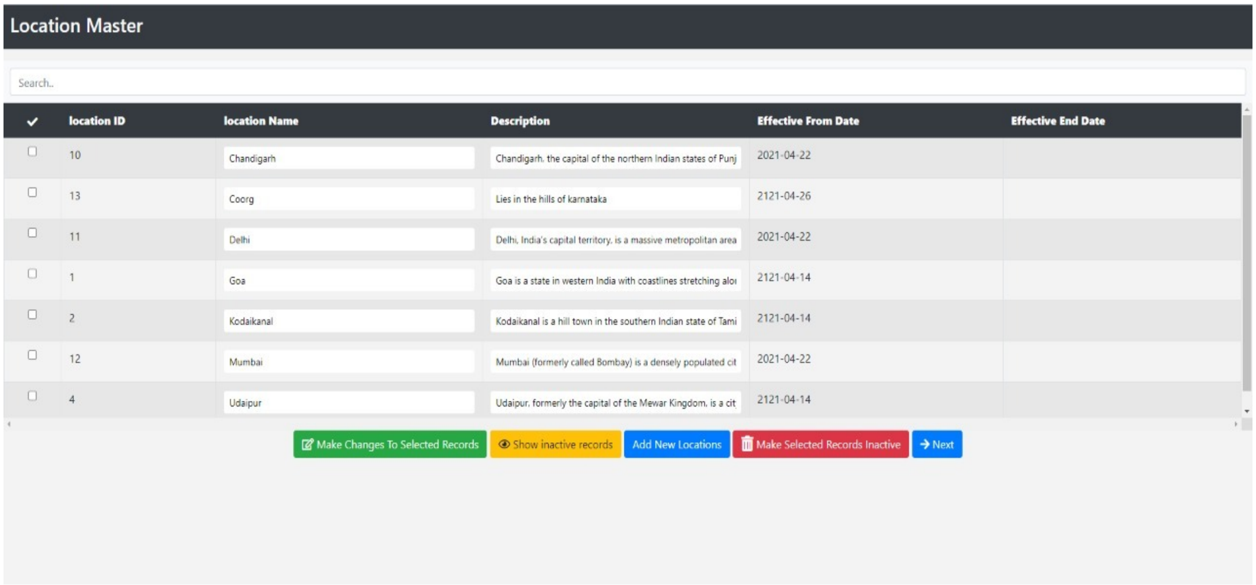
The SELECT statements are used to display all the location infos as input fields.

THE UPDATE statements are used to make chaanges to the existing records.

The INSERT statements are used to add new locations to the database.

The DELETE statements are used to delete existing records from the database.

Together, the admin panel acts as an interface between the user and the database.



Location Master

Search..

✓	location ID	location Name	Description	Effective From Date	Effective End Date
□	10	Chandigarh	Chandigarh, the capital of the northern Indian states of Punj	2021-04-22	
□	13	Coorg	Lies in the hills of karnataka	2121-04-26	
□	11	Delhi	Delhi, India's capital territory, is a massive metropolitan area	2021-04-22	
□	1	Goa	Goa is a state in western India with coastlines stretching alo	2121-04-14	
□	2	Kodaikanal	Kodaikanal is a hill town in the southern Indian state of Tami	2121-04-14	
□	12	Mumbai	Mumbai (formerly called Bombay) is a densely populated cit	2021-04-22	
□	4	Udaipur	Udaipur, formerly the capital of the Mewar Kingdom, is a cit	2121-04-14	

Make Changes To Selected Records
Show inactive records
Add New Locations
Make Selected Records Inactive
Next

The Product Design

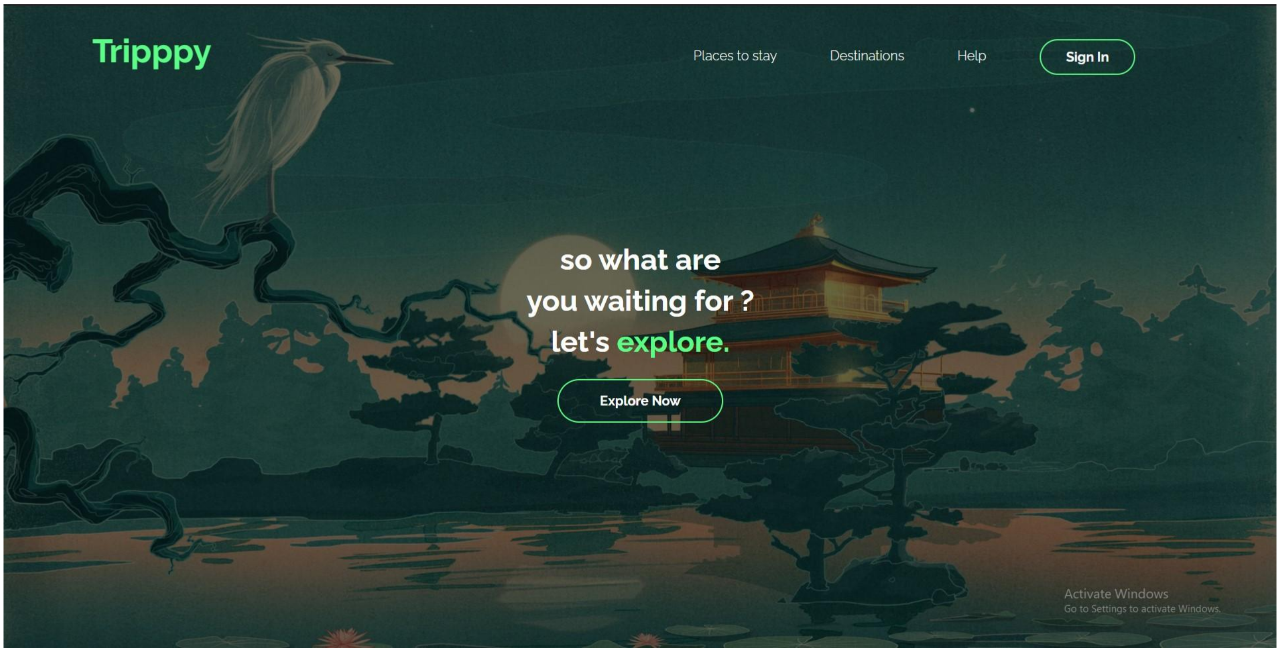
+ We have designed a vibrant and colorful UI to complement the theme and use case of the product

+ The interface is clean minimalistic and caters to the needs and wants of the user

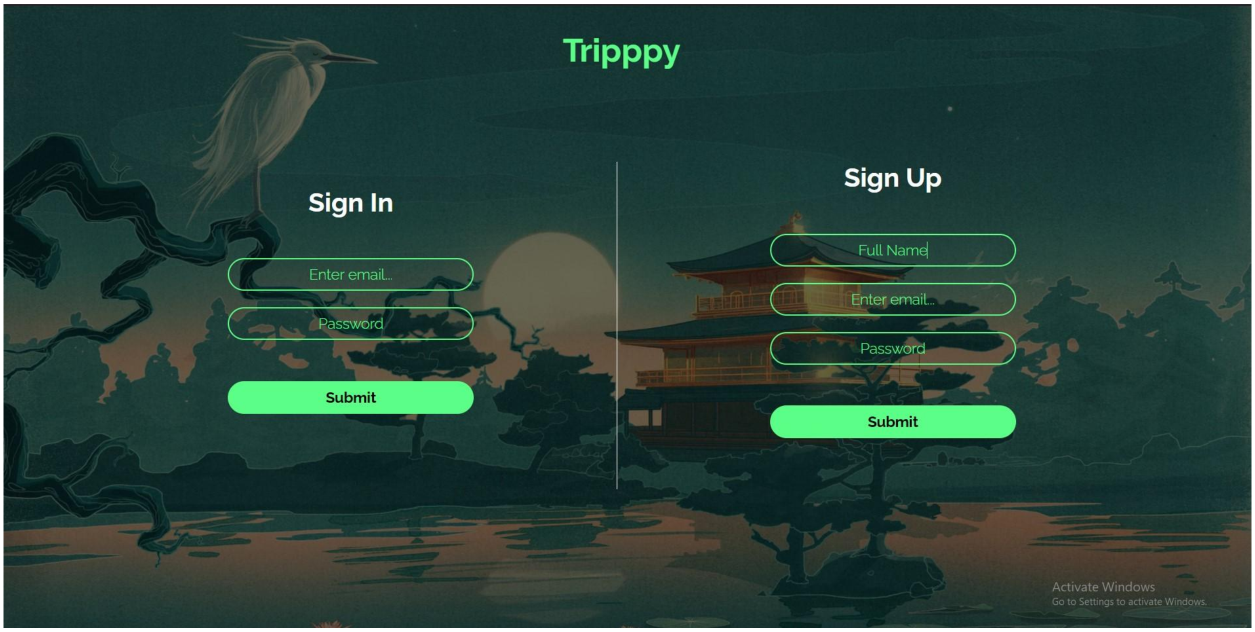
+ Thereby improves the overall experience of the user!

The User Interface

Landing Page



Sign In/Up Page

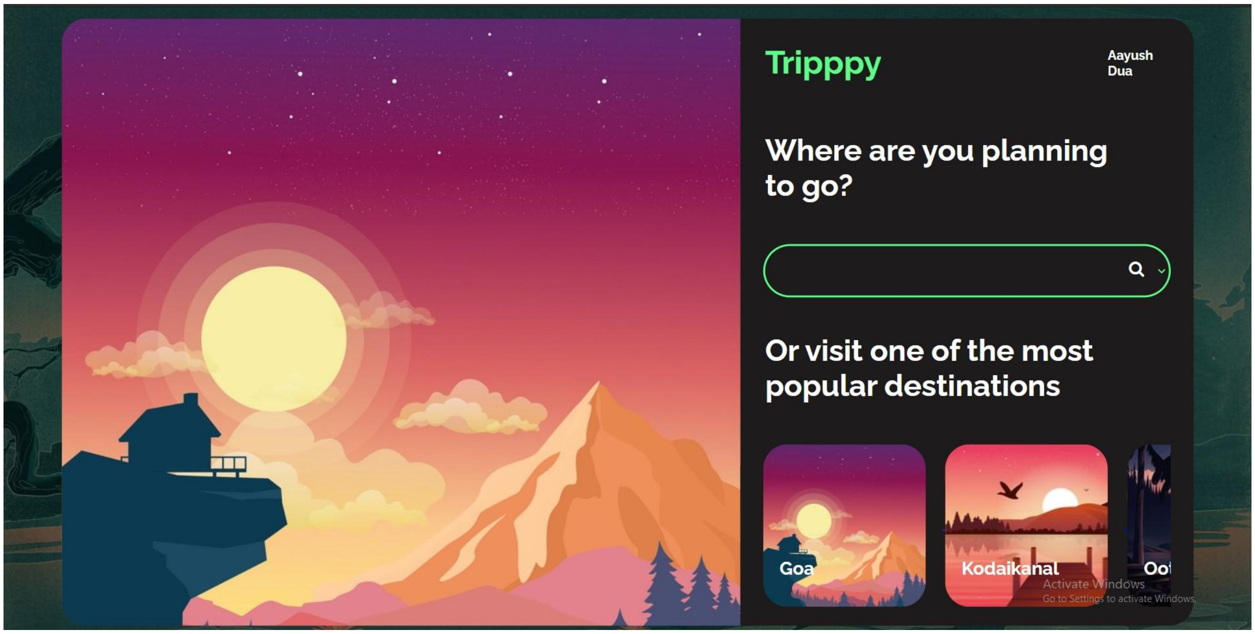


Tripppy

Sign In

Sign Up

Search Page



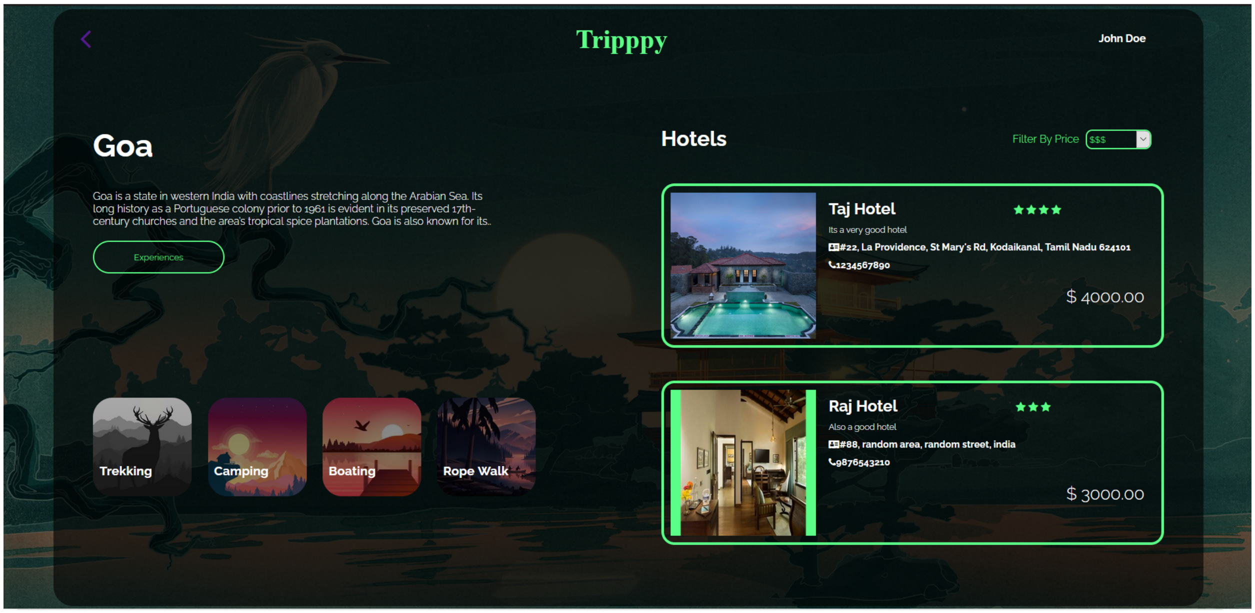
Tripppy

Aayush Dua

Where are you planning to go?

Or visit one of the most popular destinations

Display Page



Goa

Goa is a state in western India with coastlines stretching along the Arabian Sea. Its long history as a Portuguese colony prior to 1961 is evident in its preserved 17th-century churches and the area's tropical spice plantations. Goa is also known for its..

Hotels

Taj Hotel ★★★★

Its a very good hotel

#22, La Providence, St Mary's Rd, Kodaikanal, Tamil Nadu 624101

☎1234567890

$ 4000.00

Raj Hotel ★★★

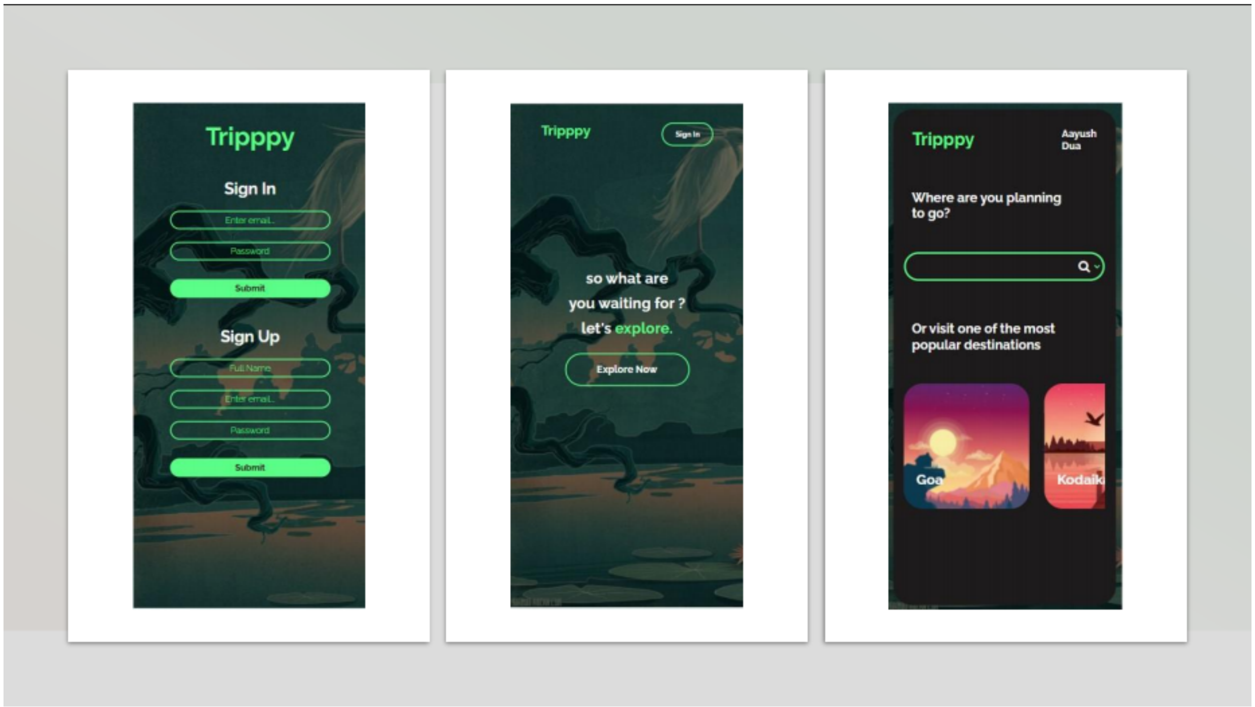
Also a good hotel

#88, random area, random street, india

☎9876543210

$ 3000.00

Mobile UI

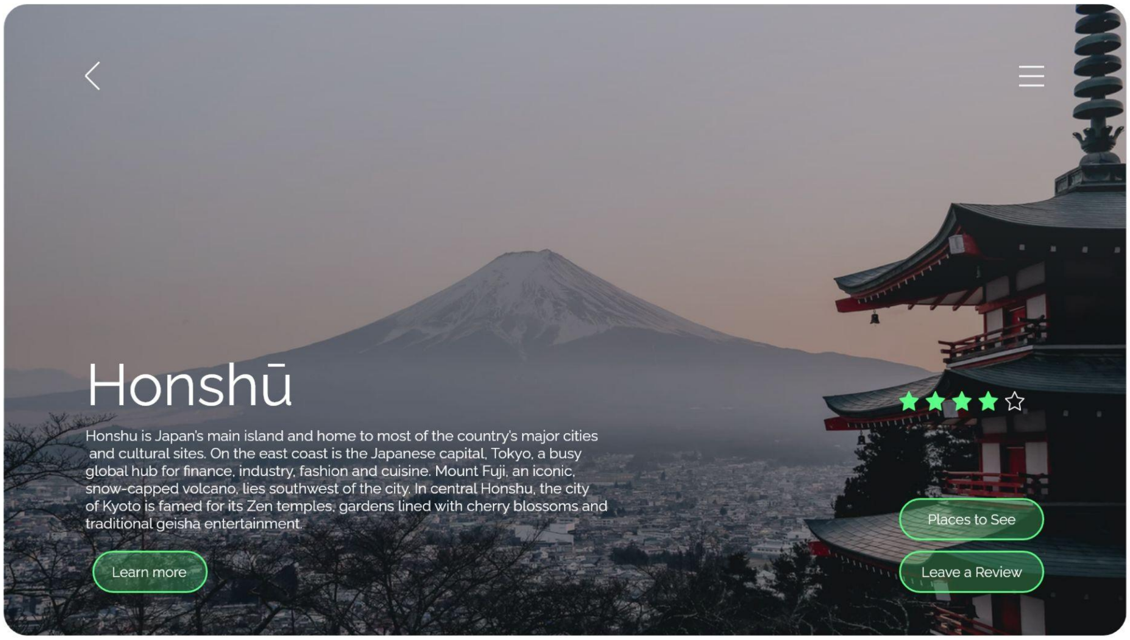


so what are you waiting for ? let's explore.

Where are you planning to go?

Or visit one of the most popular destinations

Prototypes



Honshū

Honshu is Japan's main island and home to most of the country's major cities and cultural sites. On the east coast is the Japanese capital, Tokyo, a busy global hub for finance, industry, fashion and cuisine. Mount Fuji, an iconic, snow-capped volcano, lies southwest of the city. In central Honshu, the city of Kyoto is famed for its Zen temples, gardens lined with cherry blossoms and traditional geisha entertainment.

Testing

Testing is the process of executing programs with the intention of finding out errors. During the process, the project is executed with a set of tests and the output of the website is evaluated to determine if the project is performing as expected. Testing makes a logical assumption that if all the parts of the module are correct then the goal will be successfully achieved. Testing includes after the completion of the coding phase. The project was tested from the very beginning and also at each step by entering different types of data. In the testing phase some mistakes were found, which did not come to knowledge at the time of coding the project. Then changes were made in the project coding so that it may take all the relevant data and gives the required result. All the forms were tested one by one and made all the required changes. Testing is vital to the success of the system. Testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. A small system error can conceivably explode into a much larger problem, Effective testing early in the process translates directly into long-term cost savings from a reduced number of errors. For the verification and validation of data various-nesting tasks are performed Testing is itself capable of finding the syntactical mistakes in the system but users need to test the system for logical checking.

LEVELS OF TESTING:

The aim of the testing process is to identify all the defects in the website. It is not practical to test the website with respect to each value that the input request data may assume. Testing provides a practical way of reducing defects in the website and increasing the user's confidence in a developed system. Testing consists of subjecting the website to a set of test inputs and observing the program behaves as expected. If the program fails to Testing behave as expected then conditions under which failure occurs are noted for later debugging and correction. The following things are associated with testing. Failure is a manifestation of an error. But the mere presence of an error may not necessarily lead to a failure. A test case is the triplet [I, S, O] where I am data input to the system. S is the state of the system at which the data is input, O is the expected output of the system A test suite is the set of all test cases with which a given software product is to be tested.

- **Functional Testing:**

  Here the system is a black box whose behavior is determined by studying its inputs and related outputs. The key problem is to select the inputs that have a huge probability of being members of a set in many cases; the selection of these test cases is based on the previous studies. Structural Testing: A great deal can be learnt about the strength and the limitation of the application by examining the manner in which the system breaks. This type of testing has two limitations. It tests failure behavior of the system circumstances may arise through an unexpected combination of events where the node placed on the system exceeds the maximum anticipated load. The structure of each module was checked at every step.

- **Unit Testing:**

  In unit testing the entire individual functions and modules were tested independently. By following this strategy all the errors in coding were identified and corrected. This method was applied in combination with the white and black box testing techniques to find the errors in each module. Unit testing is normally considered an adjunct to the coding step. Unit test case design was started after source level code had been developed, reviewed, and verified for correct syntax. A review of design information provides guidance for establishing test cases that were likely to uncover errors in each of the categories discussed above. Each test case was coupled with a set of expected results

Conclusion

Thus, we have completed our university project. However we wish to add more utility and purpose to our project and help it come to life. By complementing our Trip Booking application with interesting and convenient quality of life improvements based on user feedback.Therefore Feel free to give us your valuable feedback in the comments and thank you for showing interest in our project.

Bibliography

[1] PHP Manual www.php.net/

[2] https://www.mysqltutorial.org/

[3] https://www.w3schools.com/sql/

[4] https://www.siteground.com/tutorials/php-mysql/

[5] https://www.youtube.com/user/KepowOb

[6] http://www.stackoverflow.com

[7] http://www.gist.github.com

[8] https://github.com/angrbrd/trip-planner-group-project