A

PROJECT REPORT

ON

**TOUR AND TRAVELS**

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR AWARD OF THE DEGREE OF

**MASTER OF COMPUTER APPLICATIONS**

(2018-20)



**UNDER THE SUPERIVISION SUBMITTED BY**

|  |  |  |
| --- | --- | --- |
| MR. ANUPAM BHATIA  (HOD Computer Science) |  | SHUBHAM  ROLL NO: 18107  MCA 6th Sem. |

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS**

**CHAUDARY RANBIR SINGH UNIVERSITY, JIND**

**TABLE OF CONTENTS**

**Title Page No.**

Company Certificate iii

Declaration iv

Supervisor Certificate v

Acknowledgement vi

Company Profile vii

Abstract viii

List of Abbreviation ix

1. Introduction to Project 1-2

2. Objective of the Project 3

3. System Analysis 4

3.1 Software Requirement Specification 4-5

3.2 Technical Specifications 6

4. System Design 7-8

4.1 Input Design 9

4.2 Output Design 9

5. DFD and ER Diagrams 10-13

6. Site Map 14

7.Web Development 15-27

7.1 DHTML 18-20

7.2 JAVASCRIPT 21-23

7.3 CSS 24-27

8. Introduction to Python 28-39

9. Database Design 40-42

6. Snapshots 43-49

7. Implementation 50-51

7.1 Standardization of the Coding/Code Efficiency 50-56

7.2 Error Handling 52

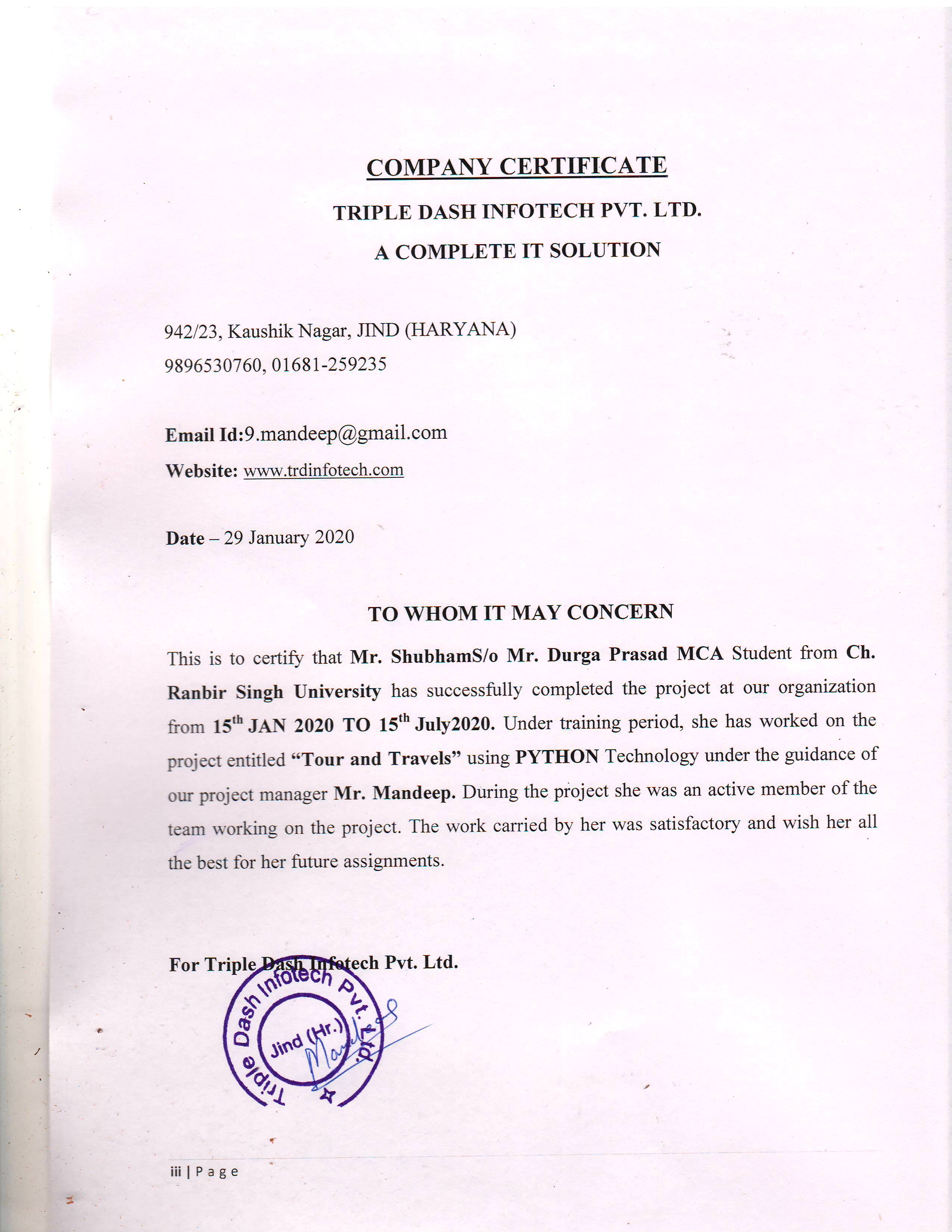
7.3 Parameters Calling/Passing 53

7.4 Validation Checks 54-56

8. Testing 57-61

8.1 Testing Strategies 58-61

9. Bibliography 62



DECLARATION

I hereby certify that the work which is being presented in this dissertation entitled “**Tour and Travels”**  in partial fulfillment of the requirement for the award of the degree of **Master of Computer Applications** and submitted to the Department of Computer Science and Applications at **Chaudhary Ranbir Singh University, JIND (HARYANA)** is an authentic record of my own work carried out during the period from January 2020 to July 2020 under the supervision of **Mr. Anupam Bhatia**, Head Of Department of Computer Science and Applications. The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute/University. If there is any copyright with this dissertation then I will be responsible for any legal dispute.

Signature of Student

**Shubham**



This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge and belief.

Signature of Supervisor

**Mr. Anupam Bhatia**

**HOD, CSA DEPARTMENT**

SUPERVISOR CERTIFICATE

This is to certify that the dissertation entitled “**TOUR AND TRAVELS”** submitted for the degree of **Master of Computer Applications** to the Department of Computer Science and Applications at **Chaudhary Ranbir Singh University, JIND (HARYANA)** during the academic year 2018-20 is Bonafede research work carried out by **“Shubham”** bearing **Roll No. ‘18107’** under my supervision and no part of this dissertation has submitted in any other University/Institute for MCA. The Work has been carried out under my supervision and this work has not been submitted elsewhere. This is further certified that she has completed all the requirements of regulations of submission of dissertation.

The viva-voce examination of **Shubham** has been held on ………………….

**(Supervisor)**

**Mr. Anupam Bhatia**

**HOD, Dept. Of CSE**

**CRSU, JIND**

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to my supervisor Mr. Anupam Bhatia, Assistant Professor & HOD, Department of Computer Science & Applications, Chaudhary Ranbir Singh University, JIND (HARYANA). I owe their overwhelming debt for helping me to plan and discuss the contents of this work. Their ideas, simulating comments, interpretations and suggestions increased my cognitive awareness and helped considerably in the fruition of my objective. I remain obliged to them for their help and able guidance through all stages of this dissertation.

I thank to all those who have contributed directly or indirectly to this work. Lastly, I would like

to thanks my parents for their years of unyielding love. They have always wanted the best for me and I admire their determination and sacrifice.

Above all, I express my indebtedness to all ALIMIGHTY for my blessing and kindness.

Shubham

COMPANY PROFILE

Like other educational and training industries, **Triple Dash Infotech Pvt. Ltd.** Offers varieties of programs but the instructors make the difference and make Triple Dash Infotech Pvt. Ltd. stand out from others. We have a variety of skilled and trained trainers whose approach is different which you can see anywhere. Ducat contributes a lot to the knowledge of its trainees and we try hard to contribute the best to increase our trainee’s ability so that they stand out from others and whatever they contribute to the corporate world automatically becomes productive. Not only the fresher but also the corporate who are not able to deal with the rising technology and software are also helped here. We try our level best to deliver our services to every corner of the world by customized education. Our motto is to deliver the best services to you and that is why we have taken the customized approach because we do not want you to compromise with your education.

ABSTRACT

The recent past showed a greater interest in recommender techniques. Now-adays there are many travel packages existing from different websites to almost all the places over the world. A customer finds it very difficult to search for the best package as he/she has to browse multiple websites, contact many travel agents and etc. which is a tedious process and is time consuming. There should be a system where the user should find the best package on the Internet with a single click. To address this issue, we adopt Travel Package Recommendation System which offers the best package among all the other packages that are on the web. This project will help tourist to suggest the best Travel Package among all the package deals on the web. On multiple demands of tourist that is, a customer will select a travel package for a particular place based on the recommendations provided by the previous customers who had experience with the package. Therefore, according to the personalized recommendations, he/she will choose the best package that is on the web. Initially, we will evaluate the particular characteristics of the current traveling packages and we mine the data on the tourists rating and the intrinsic features i.e., locations, travel seasons etc. Based on the data collected after mining, we will generate a list for personalized travel package recommendations. Furthermore, we will extract the data based on the tourist's relation with the area and season.

|  |  |
| --- | --- |
| **List of Abbreviation** | |
| Abbreviation | Full form |
| ASP | Active Server Pages |
| SQL | Structured Query Language |
| W3C | World Wide Consortium |
| API | Application Programming Interface |
| CLR | Common Language Runtime |
| HTML | Hypertext Markup Language |
| DHTML | Dynamic Hypertext Markup Language |
| CSS | Cascading Style Sheet |
| CTS | Common Type System |
| FAQ | Frequently Asked Question |
| HTTP | Hypertext Transfer Protocol |
| WWW | World Wide Web |
| XHTML | XML markup languages |
| IIS | **Internet Information Services** |

1.Introduction to Project

The Tours and Travel Management System is a web-based application. The main purpose of “Tours and travels management system” is to provide a convenient way for a customer to book hotels, flight, train and bus for 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 train, flight or bus. 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 records of the customer details of destination as well as payment received.

Tourism can be considered as most favorite pass time when people get free time. Several travel organizations are available on the web. The people or the tourist select their own Travel Package according to their personal interest. The travel companies concentrate on the interest associated with tourist making sure to increase their particular market value and supply enormous package deals. So that they can make their Travel Package more effective. Now-a-days Recommender system is becoming very famous and people are getting attracted to it, as it is helping them to choose the best package in a short time.

Recommender systems are categorized into

1. **Content based system:** With this, item recommendation is analyzed then it retrieves the information and filters this for research. For example, if the tourist goes to hill stations more often, then database contains “hill station” as recommendation.

2. **Collaborative filtering systems:** It rely on the similar factors of user or items. Preferences of different users for same item are recommended by system.

There are many challenges in designing and executing Personalized Travel Package Recommendation System.

The following shows some of the challenges:

1. The data for Travel is very less and scattered. For an example, recommendation for a movie may cost more to travel than the movie price.

2. Usually Travel package are location based so they are pertained to space or time to reach destination. For an example, the package contains locations which are geographically near and also vary season wise.

3. The older recommendation method is dependent upon rating and the travel data may not consist of this sort of rating.

2.Objective of the Project

This application is developed to provide best travelling services to the customers and travel agents. We have developed tours and travel management system to provide a search platform where a tourist can find their tour places according to their choices. This system also helps to promote responsible and interesting tourism so that people can enjoy their holidays at their favorable places. This system also helps 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 that provide healthy interaction opportunities for tourists and locals and increase better understanding of different cultures, customs, lifestyles, traditional knowledge and believe. This system also provides better way to connect with various events.  
This system also gives tours related information like which places are tourist attractions, cities, and 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. This system also keeps a history of visited places of its users.

3.System Analysis

3.1 SOFTWARE REQUIREMENT SPECIFICATION

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project. Requirements must be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirements can be architectural, Structural, functional and the nonfunctional requirement.  
The development of project needs some requirement to make the project perform better and achieves the goal of project. In developing "**Tour and Tourism**", the capabilities of computer and hardware plays a big impact on project quality. The project maker should determine the minimum requirements of hardware and also software to be used to develop a good and attractive project.

There are two phases of requirement analysis as given below:

**Primary Research:** Identifying the user requirements for making a system according to the user expectations.

**Secondary Research:** Comparing the identified requirements with already existing software having similar functionalities. Based on these researches the result is defined as the software.

Requirements analysis includes three types of activities:

**Eliciting requirements:** the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.

**Analyzing requirements:** determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.

**Recording requirements:** requirements may be documented in various forms, such

as natural-language documents, use cases, user stories, or process specifications.

Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. Historically, this has included such things as holding interviews, or holding focus groups (more aptly named in this context as requirements workshops) and creating requirements lists. More modern techniques include prototyping and use cases. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced. Systematic requirements analysis is also known as requirements engineering. it is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper, as opposed to elicitation or documentation of the requirements, for instance. Requirement engineering is a sub discipline of systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems. In some life cycle models, the requirement engineering process begins with a feasibility study activity, which leads to a feasibility report. If the feasibility study suggest that the product should be developed, then requirement analysis can begin. If requirement analysis precedes feasibility studies, which may foster outside the box thinking, then feasibility should be determined before requirements are finalize.

3.2 Technical Specifications

**Hardware Requirements:**

Processer: Intel Pentium and Above Version

Speed: 1.60 GHz

Hard Disk: 300 GB and Above

Key Board: Standard

RAM:4GB

**Software Requirements:**

Operating System: Window 7

Data Base: My SQL

Tools: HTML, CSS, JavaScript, jQuery, Python 3.7

4. SYSTEM DESIGN

System Design is the critical module in the software development after the System Analysis. Design is the technical Kernel of Software engineering. Design is a meaningful engineering representation of something. It can be traced to a customer’s requirements and at the same time accessed for quality against a set of predefined criteria for good design. It involves lot of intersection with the users; it has to be flexible and dynamic to meet the changing needs of the users over a period of time.

So, now we are in design phase. System Design is based on the requirement analysis and specification, generated in analysis phase. The design phase focuses on the detailed implementation of the system recommended in the feasibility study.

Software design is an iterative process through which requirements are translated into a ‘blueprint’ for constructing the software. Throughout the design process, the quality of the evolving design is accessed with a series of formal technical reviews or design walkthroughs.

McLaughlin [MCG 91] suggest three characteristics that serve as a guide for the evaluation of a good design-

* The design must implement all of the explicit requirements contained in the analysis model, and it must accommodate all of the implicit requirements desired by the customer.
* The design must be a readable, understandable guide for those who generate code and those who test and subsequently support the software.
* The design should provide a complete picture of the software, addressing the data, functional and behavioral domains from an implementation perspective.

Each of these characteristics is actually goal of the design process. In order to evaluate the quality of design representation, we must establish technical criteria for good design. I follow the following guidelines-

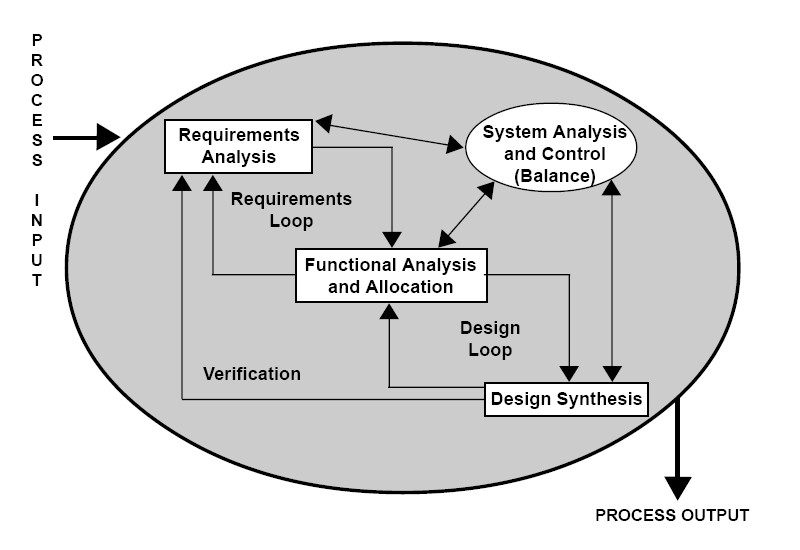
* A design should exhibit an architectural structure that:
  + Has been created using recognizable design patterns.
  + Is composed of components that exhibit good design characteristics.
  + Can be implemented in an evolutionary fashion, thereby facilitating implementation and testing.
* A design should be modular, i.e., the software should be logically portioned into elements that perform specific functions and sub functions.
* A design should contain distinct representations of data, architecture, interfaces and components (modules).
* A design should lead to data structures that are appropriate for the objects to be implemented and are drawn from recognizable data patterns.
* A design should lead to components that exhibit independent functional characteristics.
* A design should lead to interfaces that reduce the complexity of connections between modules and with the external environment.
* A design should be derived using a repeatable method that is driven by information obtained during software requirements analysis. During the system design method, we apply a set of fundamental principles and basic concepts to data, architectural, interface and component level design. Basic design principles enable to navigate the design process.
* System design is a solution “How to approach” to the creation of the new system. This important phase is composed of several steps. It provides the undertaking and procedural feasibility study. Design goes through a logical and physical stage of development. Logical design reviews present physical stage of development, prepares input/output specification and security and walkthrough.

**4.1 INPUT DESIGN**

The common cause of error in data processing inaccurate input data error entered by the data operator can be controlled by the input design. The goal of input design is to make data entry as easy, logical and free of errors as possible.

**4.2 OUTPUT DESIGN**

Computers are the most important sources of information to the users. Inputs are fed into the computers to acquire the necessary outputs. The computer can provide the well enough output in the form of information regarding various items to the users. The major form of output is the printed.



**Figure 2- Input and Output Design**

5. DFD and ER Diagrams

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system. There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements. It is generally made of symbols given below:

(1) A **square** shows the Entity: -

(2) A **Circle** shows the Process: -

(3) An **Open-Ended Rectangle** shows the data store: -

(4) An **arrow** shows the data flow: -

**Data Flow Diagram**

User Registration

Data base

User Login

User panel

Package manager

Package Booking

**ENTITY RELATIONSHIP DIAGRAM (ERD)**

**Definition -** An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

**Entity Relationship (ER) diagrams -** This diagramming technique is used to visually present a database schema or data model and was original proposed by Chen in the 1970s. There are many different data modeling notations; some are very similar to UML class diagrams (with the exception of operations). However, the notation the used here is slightly different, as proposed by Elmasri, et al.

The database schema for this system is shown in figure. The table object has been left out of the diagram because the table management feature set had been dropped from the requirements before this stage of the design process.

**E-R MODEL FOR QUERY SUBMISSION**

Feedback

Registration

TXN

Tour\_package

Customer

6.SITEMAP

7. Web Development

**HTML**

HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
* Browsers do not display the HTML tags, but use them to render the content of the page

**Html Document Structure**

<html>

<head>

Document header related tags

<.head>

<body>

Document body related tags

</body>

</html>

**A Simple HTML Document**

<!DOCTYPE html>  
<html>  
<head>  
<title>PageTitle</title>  
</head>  
<body>  
  
<h1>MyFirstHeading</h1>  
<p>Myfirstparagraph.</p>  
  
</body>  
</html>

## **HTML Tags: Most Used**

* [Heading Tag](http://http/html.com/tags/heading/)
* [Paragraph](http://http/html.com/tags/p/)
* [Italic](http://html.com/tags/i/)
* [Bold](http://html.com/tags/b/)
* [Anchor](http://html.com/tags/a/)
* [Unordered List](http://html.com/tags/ul/)
* [List Item](http://html.com/tags/li/)
* [Blockquote](http://html.com/tags/blockquote/)
* [Horizontal Rule](http://html.com/tags/hr/)
* [Image](http://html.com/tags/img/)
* [Division](http://html.com/tags/div/)

## **Formatting in html**

## **Bold Text**

Anything that appears within **<b>...</b>** element, is displayed in bold

## **Italic Text**

Anything that appears within **<i>...</i>** element is displayed in italicized.

## **Underlined Text**

Anything that appears within **<u>...</u>** element, is displayed with underline.

## **Strike Text**

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text.

* **MONOSPACED FONT**

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

## **Superscript Text**

The content of a **<sup>...</sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

## **Subscript Text**

The content of a **<sub>...</sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

## **Inserted Text**

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

## **Deleted Text**

Anything that appears within **<del>...</del>** element, is displayed as deleted text.

## **Larger Text**

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it

## **Smaller Text**

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it..

## **Grouping Content**

The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a <div> element to indicate that all of the elements within that <div> element relate to the footnotes. You might then attach a style to this <div> element so that they appear using a special set of style rules.

The <span> element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together.

7.1 DHTML

DHTML is the art of combining HTML, JavaScript, DOM, and CSS.

DHTML is NOT a language.

DHTML is a TERM describing the art of making dynamic and interactive web pages.

DHTML combines HTML, JavaScript, the HTML DOM, and CSS.

According to the World Wide Web Consortium (W3C):  
"Dynamic HTML is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be animated."

Dynamic HTML gives authors creative control so they can manipulate any page element and change styles, content and position at any time. It provides a more dynamic experience on web pages, making them more like dynamic applications and less like static content. Dynamic HTML presents richly formatted pages and lets you interact with the content on those pages without having to download additional content from the server. it means that a page can respond immediately to user actions, such as a mouse click, without having to retrieve an entire new page from the server.

With DHTML, you can easily add effects to your pages that previously were difficult to achieve. For example, you can:

* Hide content until a given time elapses or the user interacts with the page.
* Animate text and images in your document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
* Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
* Use a [**form**](https://msdn.microsoft.com/en-us/library/ms535249(v=vs.85).aspx) to capture user input, and then instantly process and respond to that data.

DHTML achieves these effects by modifying the in-memory representation of the current document and automatically reformatting it to show changes. It does not reload the document, load a new document, or require a distant server to generate new content. Instead, it uses the user's computer to calculate and carry out changes. This means a user does not wait for text and data to complete time-consuming round trips to and from a server before seeing the results. Furthermore, DHTML does not require additional support from applications or embedded controls to make changes. Typically, DHTML documents are self-contained, using styles and a script to process user input and directly manipulate the HTML elements, attributes, styles, and text of the document.

In short, DHTML eliminates the shortcomings of static pages. You can create innovative Web sites, on the Internet or on an intranet, without having to sacrifice performance for interactivity. Not only does DHTML enhance the user's perception of your documents, it also improves server performance by reducing requests to the server.

**Typically a web page using DHTML:**

<!DOCTYPE html>

<htmllang="en">

<head>

<meta charset="utf-8">

<title>DHTML example</title>

</head>

<body>

<div id="navigation"></div>

<script>

functioninit() {

varmyObj = document.getElementById("navigation");

// ... manipulate myObj

}

window.onload = init;

</script>

<!--

Often the code is stored in an external file; this is done

by linking the file that contains the JavaScript.

This is helpful when several pages use the same script:

-->

<scriptsrc="myjavascript.js"></script>

</body>

</html>

**Features of DHTML:**

1.Simplest feature is making the page dynamic.

2.Can be used to create animations, games, applications, provide new ways of navigating through web sites.

3.DHTML use low-bandwidth effect which enhance web page functionality.

4.Dynamic building of web pages is simple as no plug-in is required.  
5. Facilitates the usage of events, methods and properties and code reuse.

**Advantages of DHTML:**

1. In comparison to other multimedia software like Flash and shockwave, it took less space. Those resulted in short download time which saved bandwidth as well as made website quick.
2. DHTML was fast & zippy depending upon your device and ISP. You didn’t need an extra plugin to make it work efficiently as it used the features present in the browsers. The content loaded on the fly. The whole page didn’t load promptly and saved time for end users by altering just the content part.
3. It was supported in some form or shape by major browsers from Microsoft, Netscape. It also worked well with cell phones, PDAs, and PDFs.
4. DHTML was considered to design friendly offering web designers’ different options to give web pages a compact look. It had more advanced functionality than a static HTML & holds more content on the web page at the same time.
5. What made DHTML famous at its prime was its outlook. It was popular among clients due to its sleek nature & flexibility to make changes to their data.
6. Due to low file maintenance, it saves time for web designers and is more efficient for content management. The use of dynamic XML made the high volume of content look reasonable avoiding any lag.

**Disadvantages of DHTML:**

1. DHTML was great with functionality. However, it still required few utilities and tools that used to come at a heavy price. An example of DHTML text editor is Dreamweaver that range from $19.99 per month. Also, the training and improvement cost for transferring from HTML to DHTML makes cost rise much higher.
2. The long and complex coding structure can be daunting for few. You need to be well versed with HTML, JS, & CSS to produce a better output.
3. DHTML suffers from browser compatibility. What works for Netscape may not work on IE. If you don’t make compatibility check while writing codes, the output may be a mess.
4. It was mainly used for animation on web pages. However, due to the problem of web languages, DHTML didn’t perform too well with every platform.

7.2 JAVASCRIPT

JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari. JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

## **What is JavaScript?**

* JavaScript was designed to add interactivity to HTML pages
* JavaScript is a scripting language
* A scripting language is a lightweight programming language
* JavaScript is usually embedded directly into HTML pages
* JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
* Everyone can use JavaScript without purchasing a license

Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

## **What can a JavaScript do?**

* **JavaScript gives HTML designers a programming tool -** HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
* **JavaScript can put dynamic text into an HTML page -** A JavaScript statement like this: document. write("<h1>" + name + "</h1>") can write a variable text into an HTML page
* **JavaScript can react to events -** A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
* **JavaScript can read and write HTML elements -** A JavaScript can read and change the content of an HTML element
* **JavaScript can be used to validate data -** A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
* **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
* **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer

## **Advantages of JavaScript:**

The merits of using JavaScript are −

* **Less server interaction** – User can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − User can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** −User can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

## **Limitations of JavaScript:**

Java Script lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows us to build interactivity into otherwise static HTML pages.

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.

programmer can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script**.**

**A simple syntax of JavaScript will appear as follows:**

<html>

<head>

<script language=” JavaScript” type=”text/CSS”>

<JavaScript code>

</script>

<head>

<body>

</body>

</html>.

**The script tag takes two important attributes**

* **Language** − This attribute specifies what scripting language you are using. Typically, its value will be JavaScript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
* **Type** − This attribute is what is now recommended to indicate the scripting language in

7.3 Cascading Style Sheets (CSS):

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects. CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation. 1. CSS ─ Overview CSS 2 These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software. NOTE: The World Wide Web Consortium or W3C is a group that makes recommendations about how the Internet works and how it should evolve

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags. CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

## **Advantages of CSS :**

* **CSS saves time** − User can write CSS once and then reuse same sheet in multiple HTML pages. User can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, You do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it’s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
* **Offline Browsing** − CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
* **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

## **WHO CREATES and Maintains CSS?**

CSS was invited by **HåkonWium Lie** on October 10, 1994 and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called **specifications**. When a specification has been discussed and officially ratified by W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

## **CSS Versions**

Cascading Style Sheets, level 1 (CSS1) was came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 was became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

CSS3 was became a W3C recommendation in June 1999 and builds on older versions CSS. it has divided into documentations is called as Modules and here each module having new extension features defined in CSS2.

### **CSS3 Modules**

CSS3 Modules are having old CSS specifications as well as extension features.

* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
* **Property** - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
* **Value** - Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.



* use and its value should be set to "text/JavaScript".

INTRODUCTION TO PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## **Python Features**

Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Python ─ Environment**

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**Local Environment Setup**

Open a terminal window and type "python" to find out if it is already installed and which version is installed.

* Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
* Macintosh (Intel, PPC, 68K)
* OS/2
* DOS (multiple versions)
* PalmOS
* Nokia mobile phones
* Windows CE
* Acorn/RISC OS
* BeOS
* Amiga
* VMS/OpenVMS
* QNX
* VxWorks
* Python has also been ported to the Java and .NET virtual machines

**Getting Python**

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python: http://www.python.org/.

You can download Python documentation from www.python.org/doc/. The documentation is available in HTML, PDF, and PostScript formats.

**Installing Python**

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python. If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms:

**Unix and Linux Installation**

Here are the simple steps to install Python on Unix/Linux machine.

 Open a Web browser and go to http://www.python.org/download/.

 Follow the link to download zipped source code available for Unix/Linux.

 Download and extract files.

 Editing the Modules/Setupfile if you want to customize some options.

 **run.** /configure script

 make

 make install

This installs Python at standard location /usr/local/binand its libraries at /usr/local/lib/pythonXXwhere XX is the version of Python.

**Windows Installation**

Here are the steps to install Python on Windows machine.

 Open a Web browser and go to http://www.python.org/download/

 Follow the link for the Windows installer python-XYZ.*msi* file where XYZ is the version you need to install.

 To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

 Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

**Setting up PATH**

* Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.
* The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.
* The **path** variable is named as PATH in Unix or Path in Windows (Unix is case-sensitive; Windows is not).
* In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.

**Setting path at Unix/Linux**

To add the Python directory to the path for a particular session in Unix:

* **In the csh shell:** type setenv PATH "$PATH:/usr/local/bin/python" and press Enter.
* **In the bash shell (Linux):** type export ATH="$PATH:/usr/local/bin/python" and press Enter.
* **In the sh or ksh shell:** type PATH="$PATH:/usr/local/bin/python" and press Enter.

**Note:** /usr/local/bin/python is the path of the Python directory

**Setting path at Windows**

To add the Python directory to the path for a particular session in Windows:

* **At the command prompt:** type path %path%; C:\Python and press Enter.

**Note:** C:\Python is the path of the Python directory.

**Python Environment Variables**

Here are important environment variables, which can be recognized by Python:

|  |  |
| --- | --- |
| **Variable** | **Description** |
| PYTHONPATH | It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. Python Path is sometimes preset by Python installer. | |
| PYTHONSTARTUP | It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH. | |
| PYTHONCASEOK | It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it. | |
| PYTHONHOME | It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy. | |

**Running Python**

There are three different ways to start Python:

**(1) Interactive Interpreter**

You can start Python from Unix, DOS, or any other system that provides you a command-line interpreter or shell window.

Enter **python** the command line.

Start coding right away in the interactive interpreter.

$python # Unix/Linux

or

python% # Unix/Linux

or

C:>python # Windows/DOS

Here is the list of all the available command line options:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Option** | **Description** | |  |  | | -d | It provides debug output. | | -O | It generates optimized bytecode (resulting in .pyo files). | | -S | Do not run import site to look for Python paths on startup. | | -v | verbose output (detailed trace on import statements). | | -X | disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6. | | -c cmd | run Python script sent in as cmd string | | file | run Python script from given file | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**(2) Script from the Command-line**

A Python script can be executed at command line by invoking the interpreter on your application, as in the following:

$python script.py # Unix/Linux

Or

python% script.py # Unix/Linux

or

C:>python script.py # Windows/DOS

**Note:** Be sure the file permission mode allows execution.

**(3) Integrated Development Environment**

You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

 **Unix:** IDLE is the very first Unix IDE for Python.

 **Windows:** Python Win is the first Windows interface for Python and is an IDE with a GUI.

 **Macintosh:** The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.

If you are not able to set up the environment properly, then you can take help from your system admin. Make sure the Python environment is properly set up and working perfectly fine.

**Note:** All the examples given in subsequent chapters are executed with Python 2.4.3 version available on CentOS flavor of Linux.

**Data Types In Python**

Number data types store numeric values. They are immutable data types, means that changing the value of a number data type results in a newly allocated object.

Number objects are created when you assign a value to them. For example −

var1 = 1

var2 = 10

You can also delete the reference to a number object by using the **del** statement. The syntax of the del statement is −

del var1[, var2[, var3[...., varN]]]]

You can delete a single object or multiple objects by using the **del** statement. For example −

del var

del var\_a, var\_b

Python supports four different numerical types −

* **int (signed integers)** − They are often called just integers or ints, are positive or negative whole numbers with no decimal point.
* **long (long integers)** − Also called longs, they are integers of unlimited size, written like integers and followed by an uppercase or lowercase L.
* **float (floating point real values)** − Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10 (2.5e2 = 2.5 x 102 = 250).
* **complex (complex numbers)** − are of the form a + bJ, where a and b are floats and J (or j) represents the square root of -1 (which is an imaginary number). The real part of the number is a, and the imaginary part is b. Complex numbers are not used much in Python programming.

Examples

Here are some examples of numbers

|  |  |  |  |
| --- | --- | --- | --- |
| **Int** | **long** | **float** | **Complex** |
| 10 | 51924361L | 0.0 | 3.14j |
| 100 | -0x19323L | 15.20 | 45.j |
| -786 | 0122L | -21.9 | 9.322e-36j |
| 080 | 0xDEFABCECBDAECBFBAEL | 32.3+e18 | .876j |
| -0490 | 535633629843L | -90. | -.6545+0J |
| -0x260 | -052318172735L | -32.54e100 | 3e+26J |
| 0x69 | -4721885298529L | 70.2-E12 | 4.53e-7j |

* Python allows you to use a lowercase L with long, but it is recommended that you use only an uppercase L to avoid confusion with the number 1. Python displays long integers with an uppercase L.
* A complex number consists of an ordered pair of real floating-point numbers denoted by a + bj, where a is the real part and b is the imaginary part of the complex number.

Number Type Conversion

Python converts numbers internally in an expression containing mixed types to a common type for evaluation. But sometimes, you need to coerce a number explicitly from one type to another to satisfy the requirements of an operator or function parameter.

* Type **int(x)** to convert x to a plain integer.
* Type **long(x)** to convert x to a long integer.
* Type **float(x)** to convert x to a floating-point number.
* Type **complex(x)** to convert x to a complex number with real part x and imaginary part zero.
* Type **complex(x, y)** to convert x and y to a complex number with real part x and imaginary part y. x and y are numeric expressions

Mathematical Functions

Python includes following functions that perform mathematical calculations.

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Returns ( description )** |
| 1 | [**abs(x)**](https://www.tutorialspoint.com/python/number_abs.htm)  The absolute value of x: the (positive) distance between x and zero. |
| 2 | [**ceil(x)**](https://www.tutorialspoint.com/python/number_ceil.htm)  The ceiling of x: the smallest integer not less than x |
| 3 | [**cmp(x, y)**](https://www.tutorialspoint.com/python/number_cmp.htm)  -1 if x < y, 0 if x == y, or 1 if x > y |
| 4 | [**exp(x)**](https://www.tutorialspoint.com/python/number_exp.htm)  The exponential of x: ex |
| 5 | [**fabs(x)**](https://www.tutorialspoint.com/python/number_fabs.htm)  The absolute value of x. |
| 6 | [**floor(x)**](https://www.tutorialspoint.com/python/number_floor.htm)  The floor of x: the largest integer not greater than x |
| 7 | [**log(x)**](https://www.tutorialspoint.com/python/number_log.htm)  The natural logarithm of x, for x> 0 |
| 8 | [**log10(x)**](https://www.tutorialspoint.com/python/number_log10.htm)  The base-10 logarithm of x for x> 0. |
| 9 | [**max(x1, x2,...)**](https://www.tutorialspoint.com/python/number_max.htm)  The largest of its arguments: the value closest to positive infinity |
| 10 | [**min(x1, x2,...)**](https://www.tutorialspoint.com/python/number_min.htm)  The smallest of its arguments: the value closest to negative infinity |
| 11 | [**modf(x)**](https://www.tutorialspoint.com/python/number_modf.htm)  The fractional and integer parts of x in a two-item tuple. Both parts have the same sign as x. The integer part is returned as a float. |
| 12 | [**pow(x, y)**](https://www.tutorialspoint.com/python/number_pow.htm)  The value of x\*\*y. |
| 13 | [**round(x [,n])**](https://www.tutorialspoint.com/python/number_round.htm)  **x** rounded to n digits from the decimal point. Python rounds away from zero as a tie-breaker: round (0.5) is 1.0 and round (-0.5) is -1.0. |
| 14 | [**sqrt(x)**](https://www.tutorialspoint.com/python/number_sqrt.htm)  The square root of x for x > 0 |

**Random Number Functions**

Random numbers are used for games, simulations, testing, security, and privacy applications. Python includes following functions that are commonly used.

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Description** |
| 1 | [**choice(seq)**](https://www.tutorialspoint.com/python/number_choice.htm)  A random item from a list, tuple, or string. |
| 2 | [**randrange ([start,] stop [,step])**](https://www.tutorialspoint.com/python/number_randrange.htm)  A randomly selected element from range(start, stop, step) |
| 3 | [**random()**](https://www.tutorialspoint.com/python/number_random.htm)  A random float r, such that 0 is less than or equal to r and r is less than 1 |
| 4 | [**seed([x])**](https://www.tutorialspoint.com/python/number_seed.htm)  Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None. |
| 5 | [**shuffle(lst)**](https://www.tutorialspoint.com/python/number_shuffle.htm)  Randomizes the items of a list in place. Returns None. |
| 6 | [**uniform(x, y)**](https://www.tutorialspoint.com/python/number_uniform.htm)  A random float r, such that x is less than or equal to r and r is less than y |

**Trigonometric Functions**

Python includes following functions that perform trigonometric calculations.

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Description** |
| 1 | [**acos(x)**](https://www.tutorialspoint.com/python/number_acos.htm)  Return the arc cosine of x, in radians. |
| 2 | [**asin(x)**](https://www.tutorialspoint.com/python/number_asin.htm)  Return the arc sine of x, in radians. |
| 3 | [**atan(x)**](https://www.tutorialspoint.com/python/number_atan.htm)  Return the arc tangent of x, in radians. |
| 4 | [**atan2(y, x)**](https://www.tutorialspoint.com/python/number_atan2.htm)  Return atan(y / x), in radians. |
| 5 | [**cos(x)**](https://www.tutorialspoint.com/python/number_cos.htm)  Return the cosine of x radians. |
| 6 | [**hypot(x, y)**](https://www.tutorialspoint.com/python/number_hypot.htm)  Return the Euclidean norm, sqrt(x\*x + y\*y). |
| 7 | [**sin(x)**](https://www.tutorialspoint.com/python/number_sin.htm)  Return the sine of x radians. |
| 8 | [**tan(x)**](https://www.tutorialspoint.com/python/number_tan.htm)  Return the tangent of x radians. |
| 9 | [**degrees(x)**](https://www.tutorialspoint.com/python/number_degrees.htm)  Converts angle x from radians to degrees. |
| 10 | [**radians(x)**](https://www.tutorialspoint.com/python/number_radians.htm)  Converts angle x from degrees to radians. |

Mathematical Constants

The module also defines two mathematical constants −

|  |  |
| --- | --- |
| **Sr.No.** | **Constants & Description** |
| 1 | **Pi**  The mathematical constant pi. |
| 2 | **E**  The mathematical constant e. |

Database Design

**Sql query**

create table Customer (Customer\_Id int primary key, Name varchar (20), Age varchar (3), Contact\_no varchar (50), Address varchar (50), Email\_id varchar (30), City varchar (20), password varchar (20))

|  |  |
| --- | --- |
| Customer\_Id | Int (Primary key) |
| Name | varchar (20) |
| Age | varchar (3) |
| Contact\_no | varchar (50) |
| Address | varchar (50) |
| Email\_id | varchar (30) |
| City | varchar (20) |
| Password | varchar (20) |

**Sql query**

create table Tour\_package (Package\_Id int primary key,Package\_Name varchar (20),Description varchar (50),Cost int,Discount int)

|  |  |
| --- | --- |
| Package\_Id | Int (Primary key) |
| Package\_Name | varchar (20) |
| Description | varchar (50) |
| Cost | Int |
| Discount | Int |

**Sql Query**

create table Registration (Reg\_Id int primary key,Customer\_Id int foreign key references customer(customer\_id),Package\_Id int foreign key references Tour\_package(package\_Id),date,No\_of\_person int,initial\_deposit int,remark varchar (50))

|  |  |
| --- | --- |
| Reg\_Id | Int (Primary key) |
| Customer\_Id | Int (foreign key) |
| Package\_Id | Int (foreign key) |
| date | Date |
| No\_of\_person | Int |
| initial\_deposit | Int |
| Remark | varchar (50) |

**Sql Query**

create table Txn (Txn\_Id int primary key,Customer\_Id int foreign key references customer(customer\_id),Package\_Id int foreign key references Tour\_package(package\_Id),Amount int,date,remark varchar (500))

|  |  |
| --- | --- |
| Txn\_Id | Int (Primary key) |
| Customer\_Id | Int (foreign key) |
| Package\_Id | Int (foreign key) |
| Amount | Int |
| Date | Date |
| Remark | varchar (500) |

**Sql Query**

create table feedback (feedback\_Id int primary key, Customer\_Id int foreign key references customer(customer\_id), Package\_Id int foreign key references Tour\_package(package\_Id), feedback varchar (500), date, Rating int,)

|  |  |
| --- | --- |
| feedback\_Id | Int (Primary key) |
| Customer\_Id | Int (foreign key) |
| Package\_Id | Int (foreign key) |
| feedback | Varchar (500) |
| date | Date |
| Rating | int |

6. Snapshots

**Implementation**

.

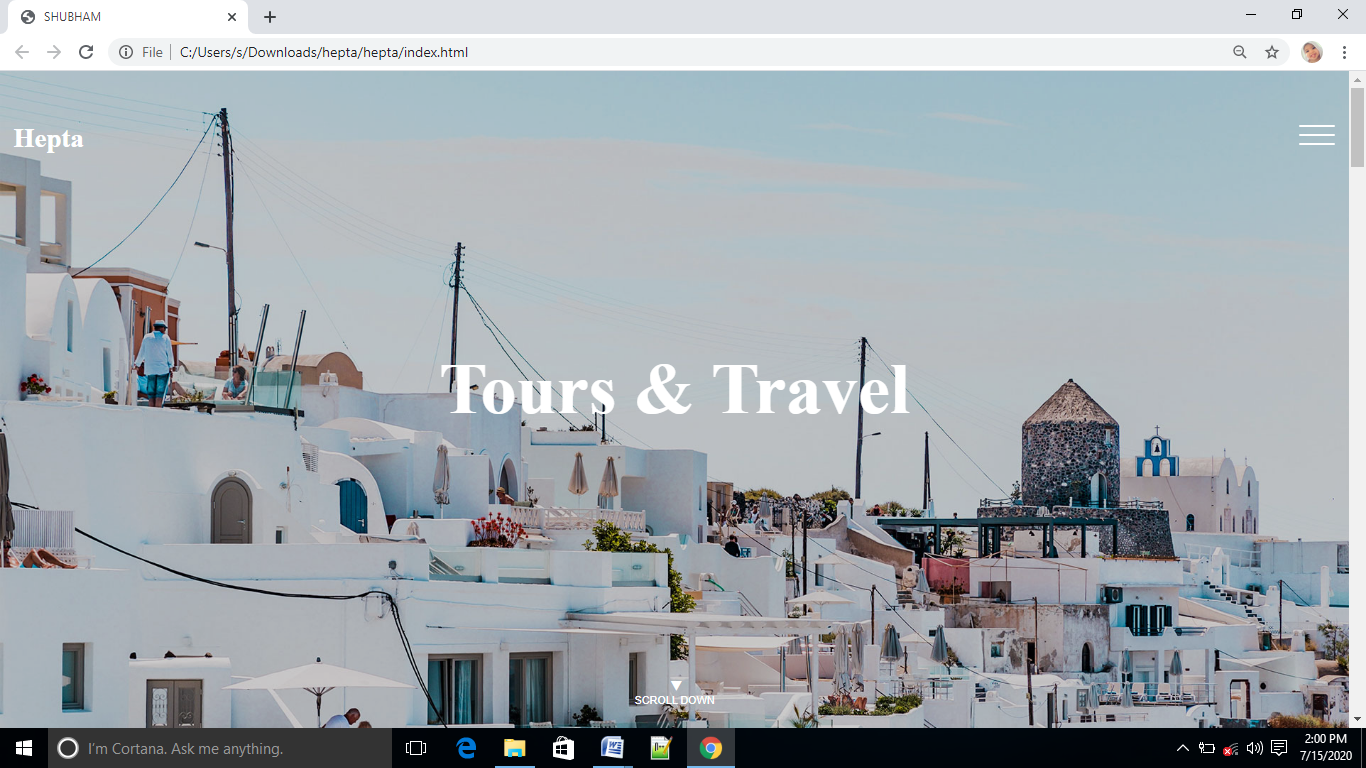


Figure Home page

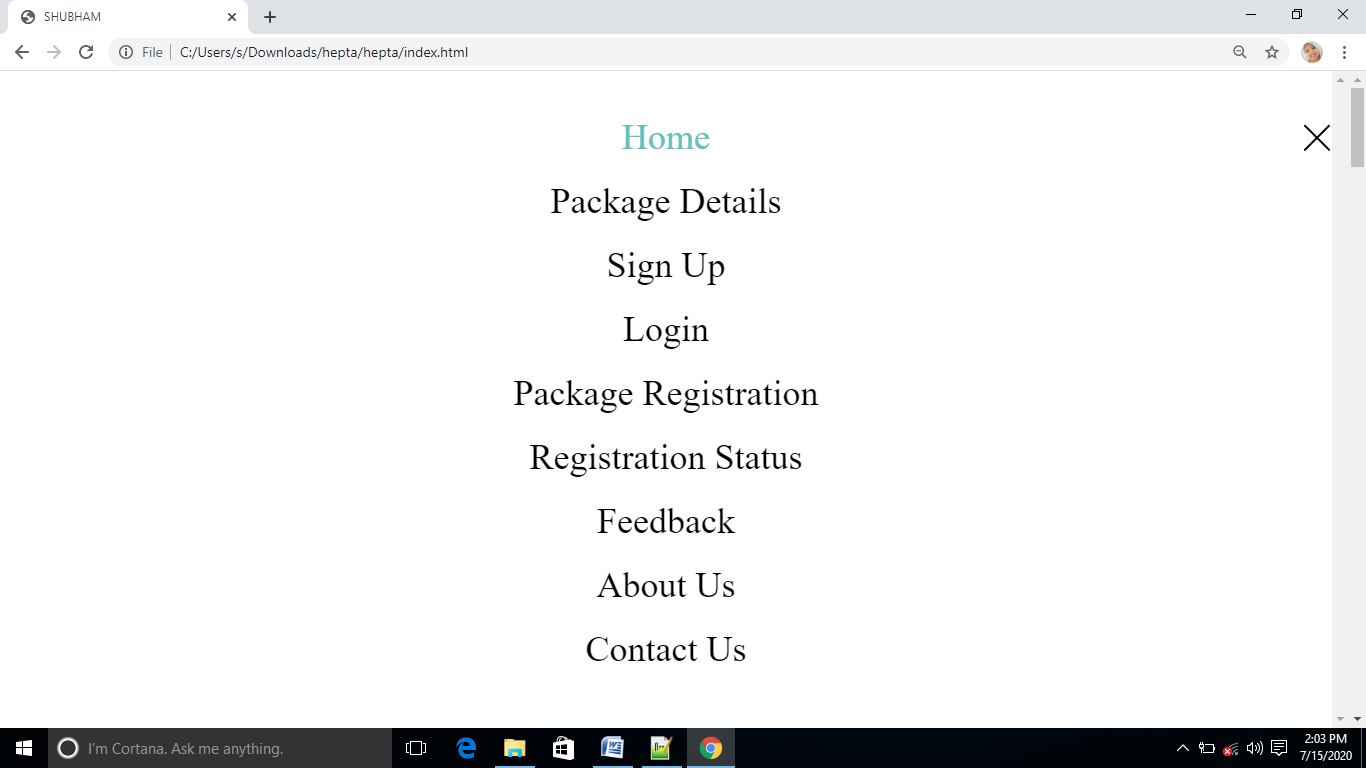


Figure Menu

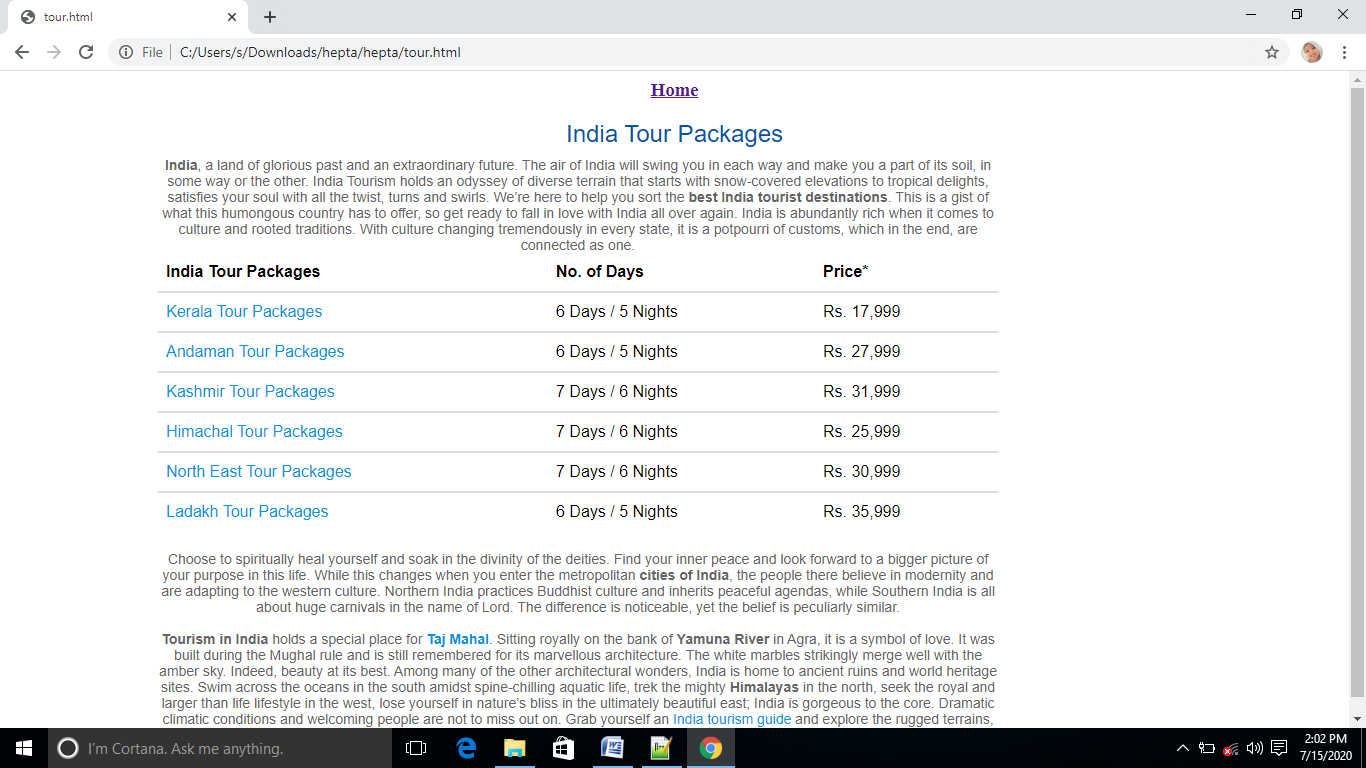


Figure Tour Package

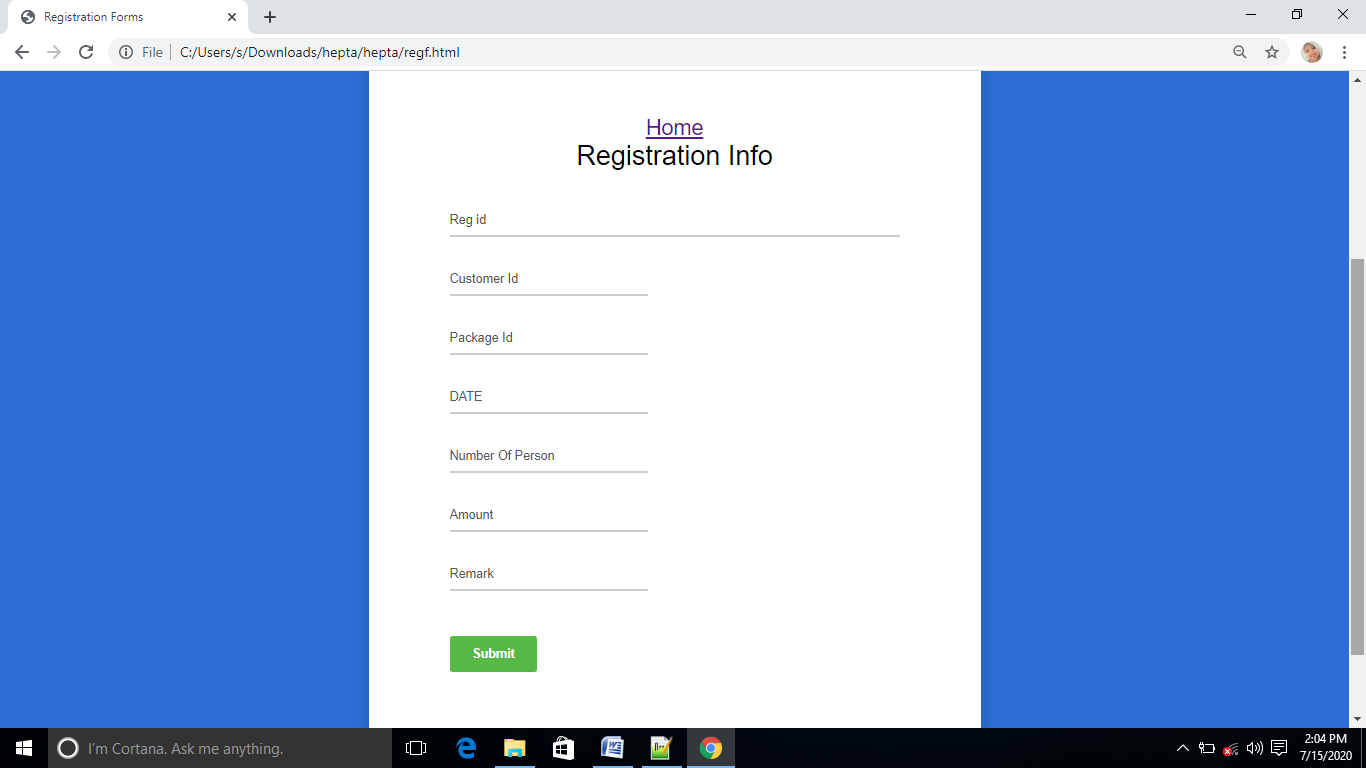


Figure Registration

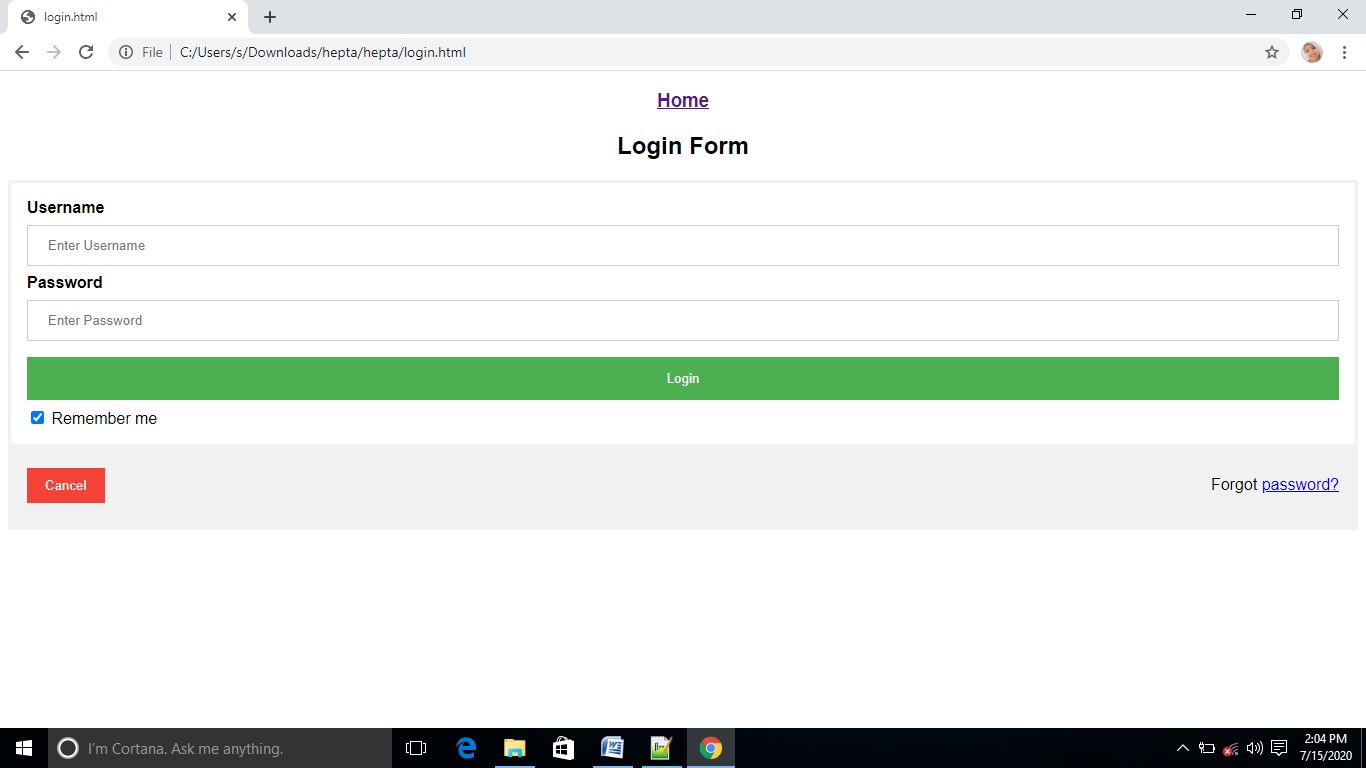


Figure Login Page

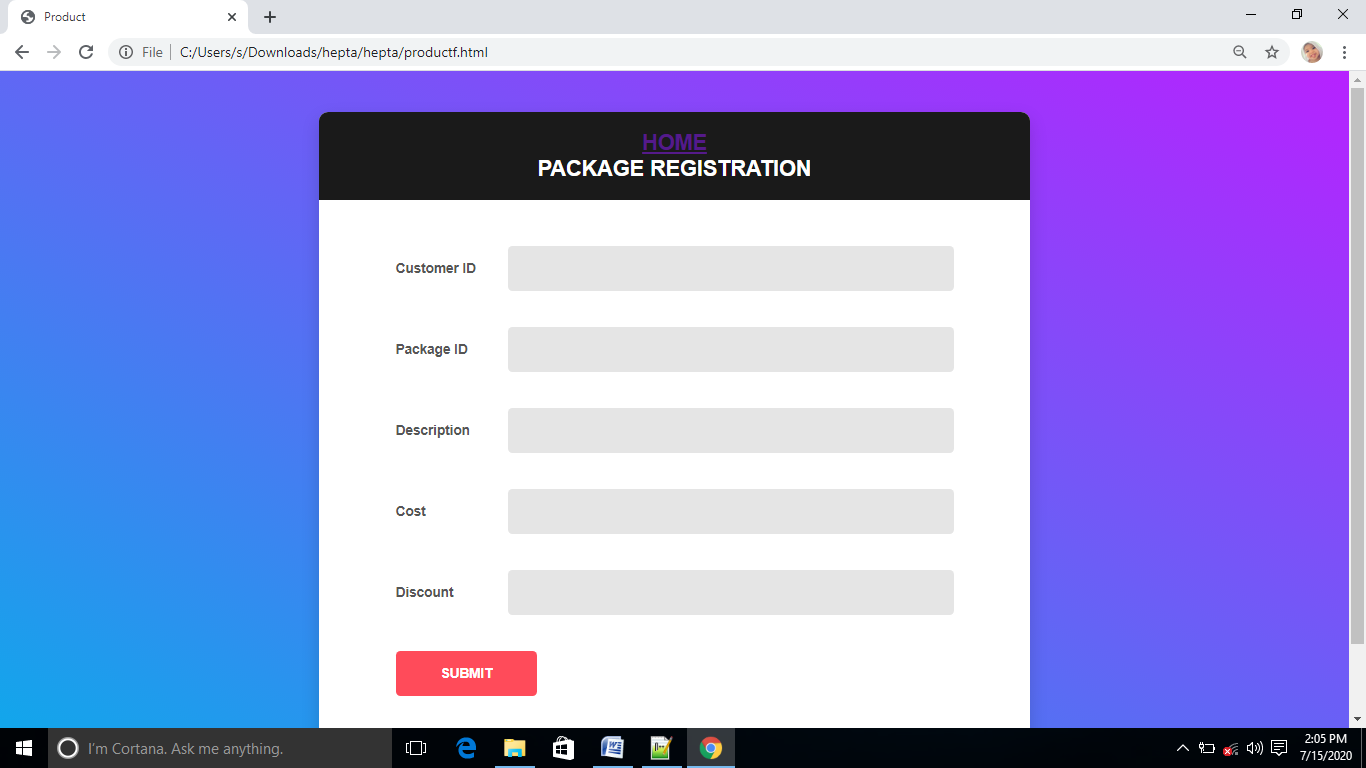


Figure Package Registration

Figure Customer Form

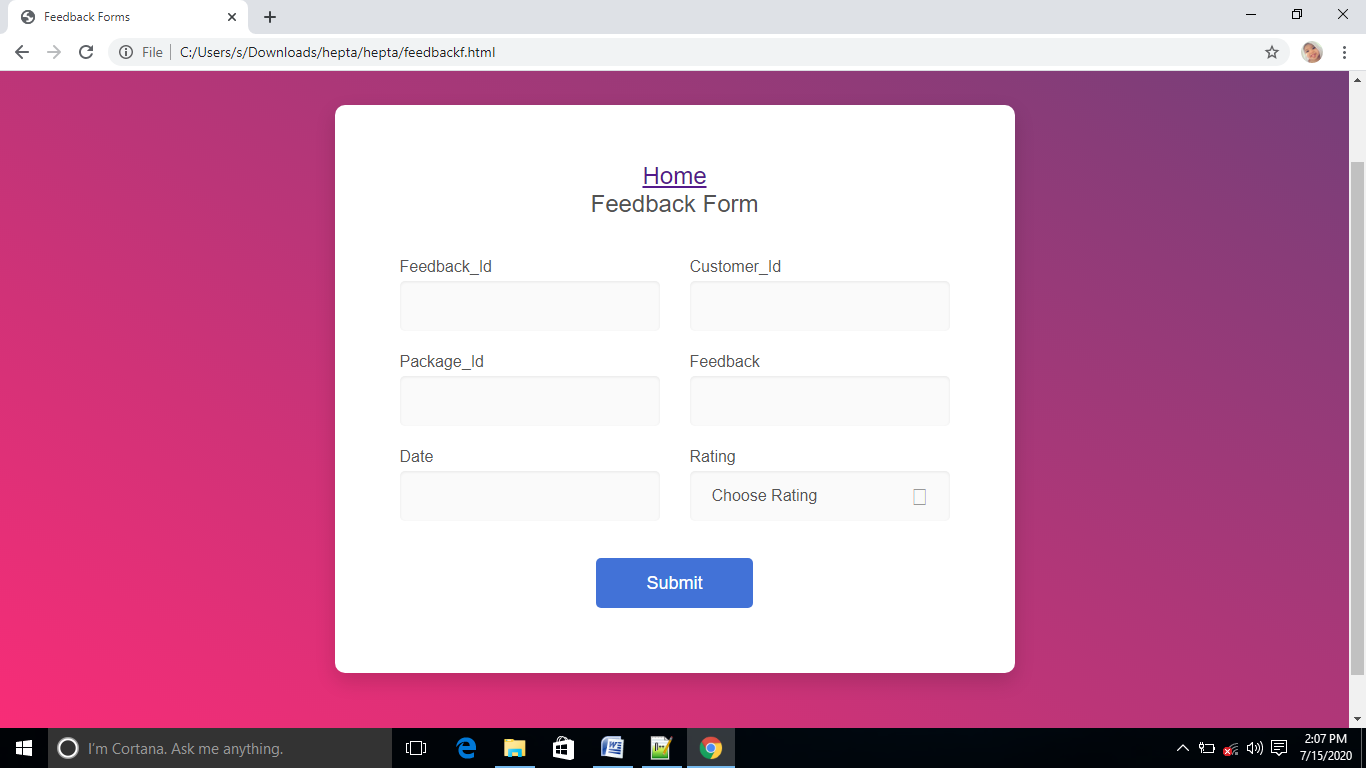


Figure Feedback form

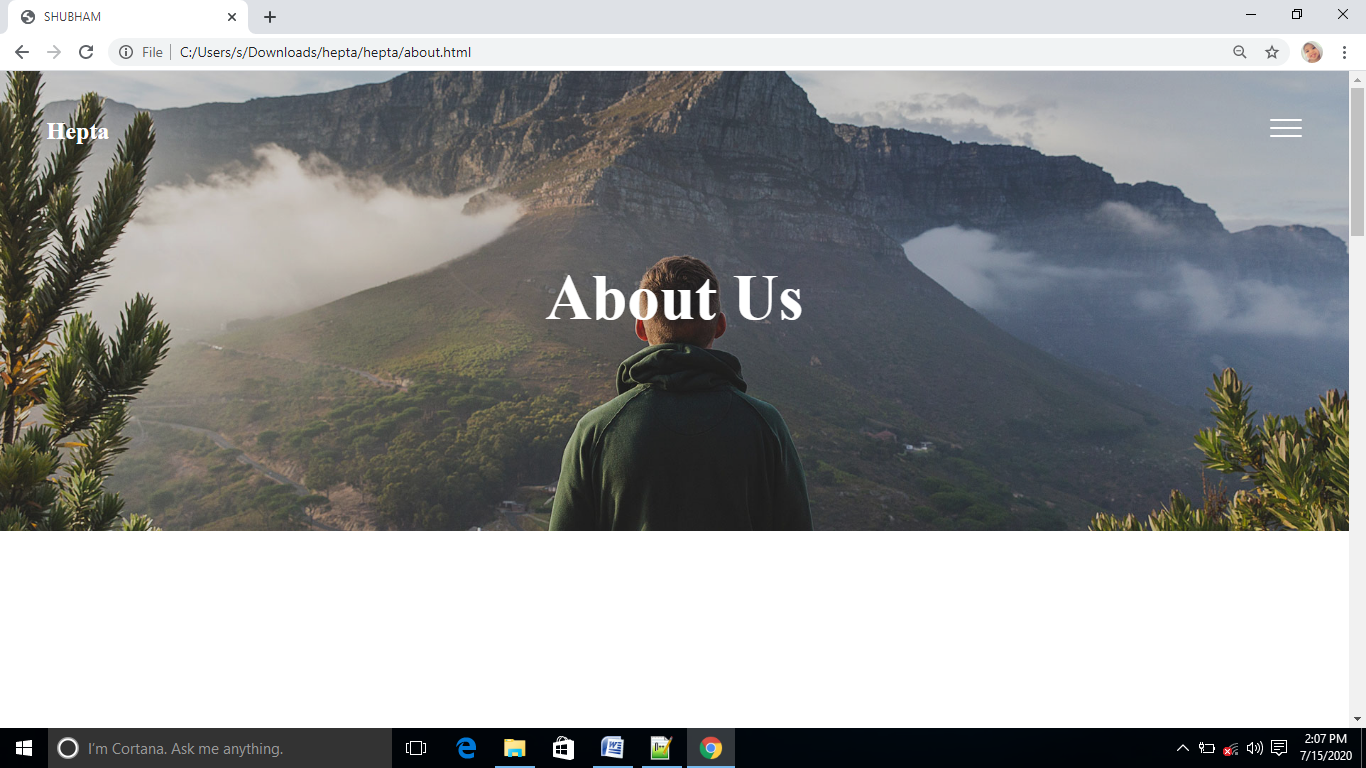
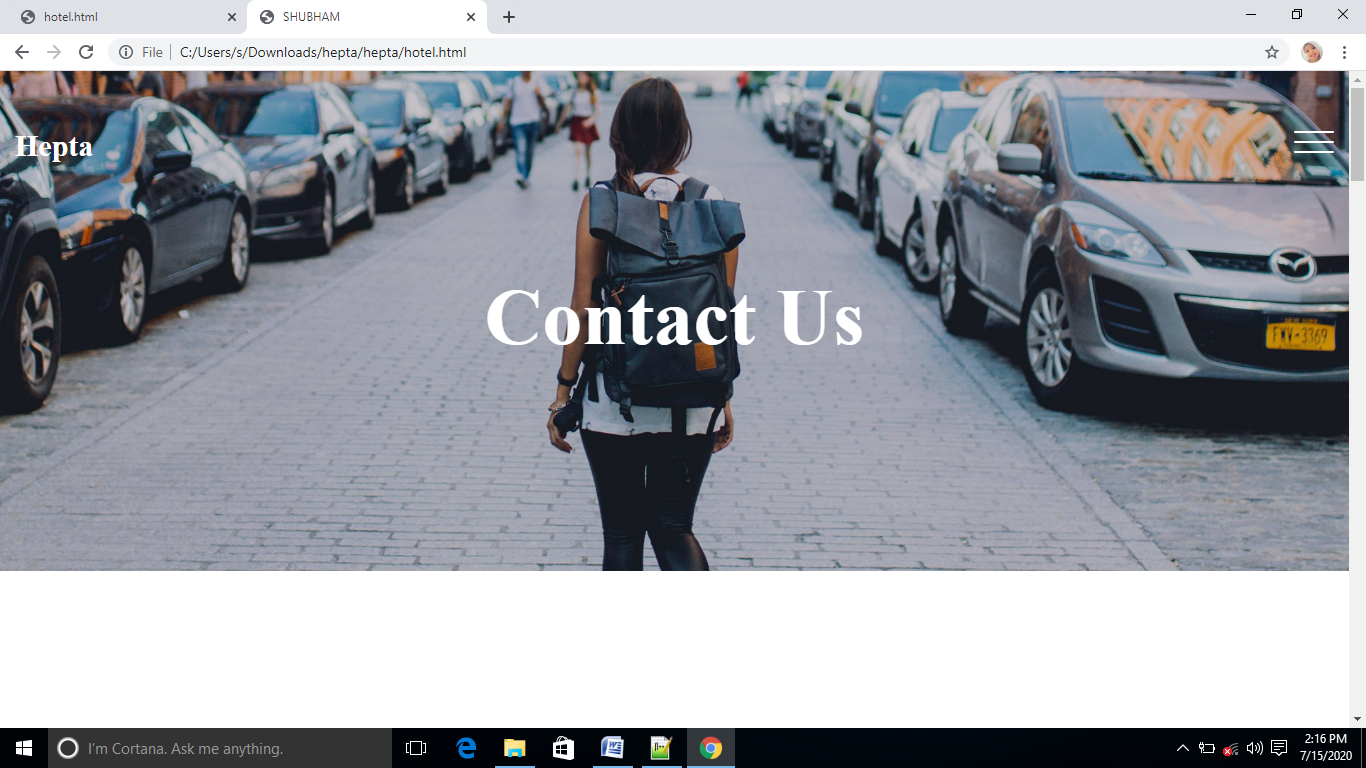


Figure About us

Figure Contact us

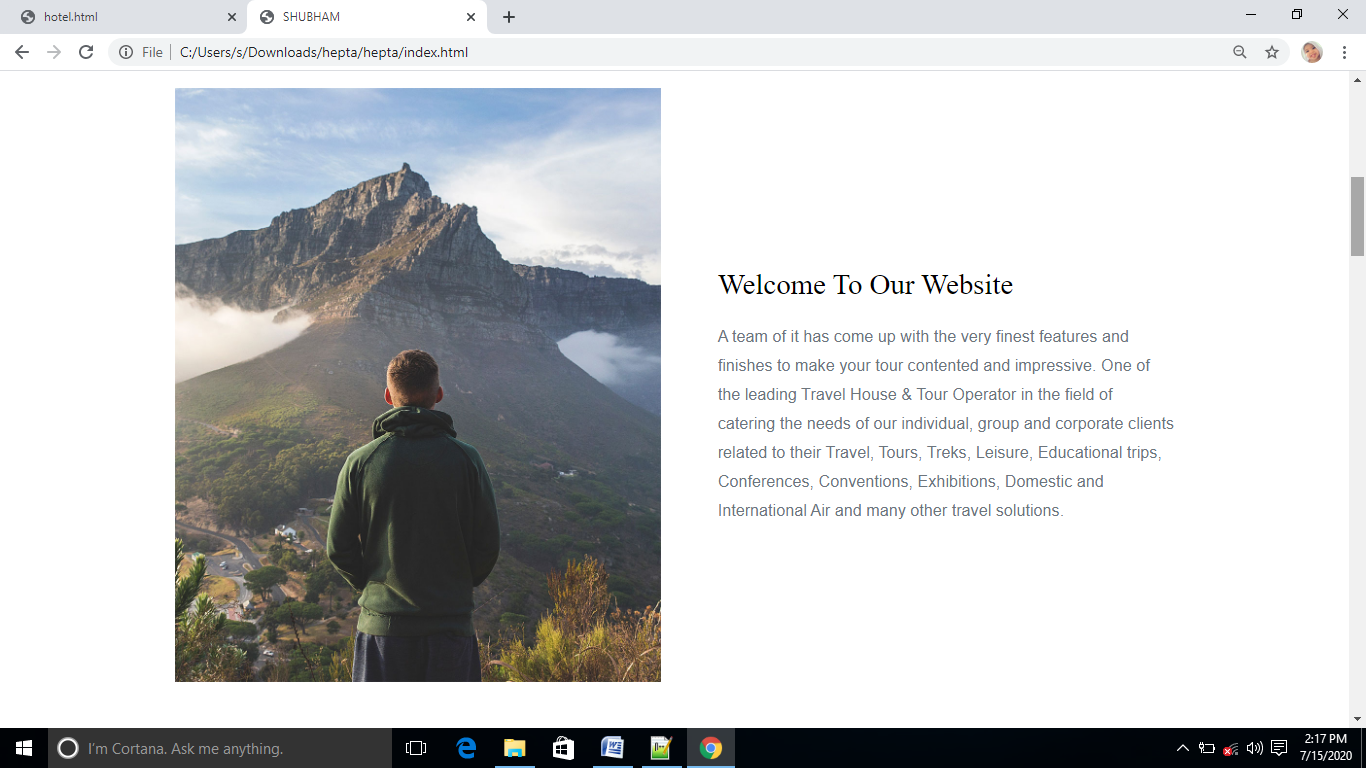
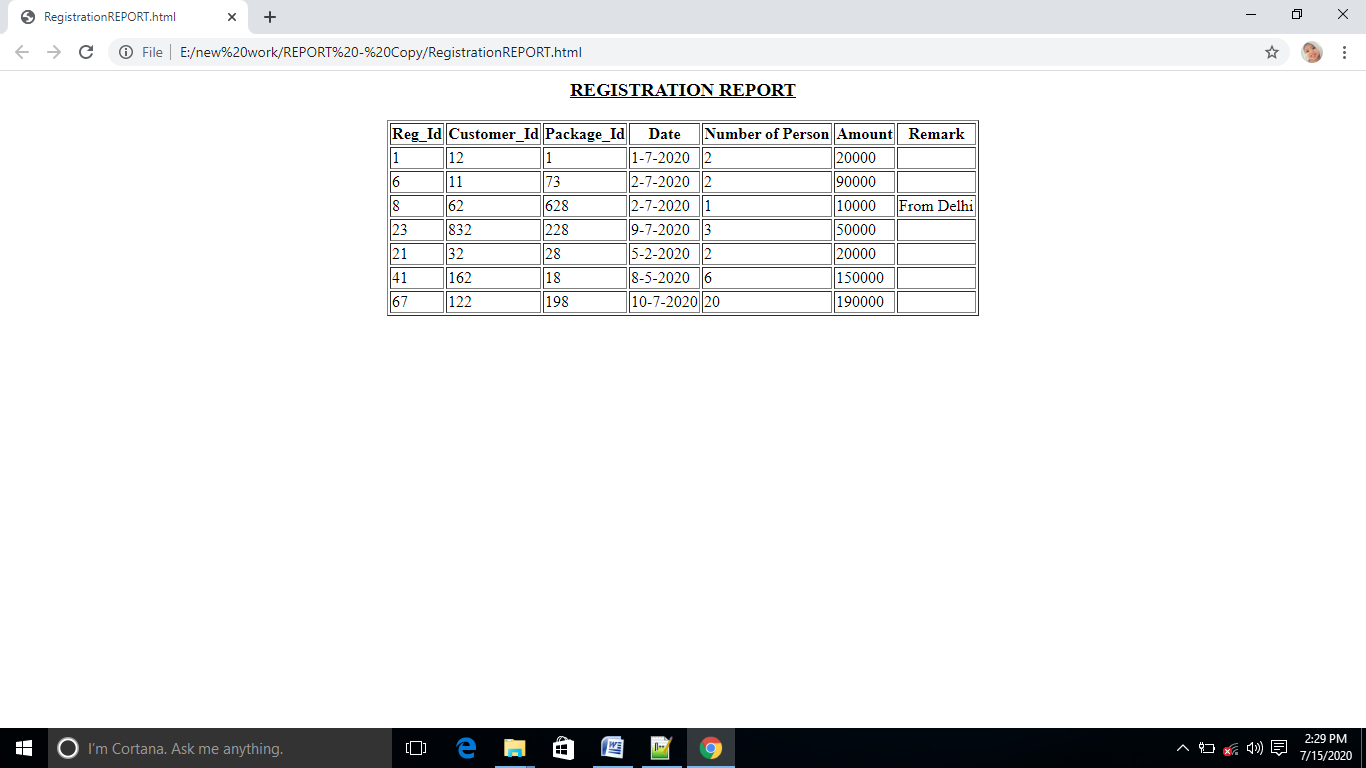


Figure Welcome to our website



### Figure REGISTRATION REPORT

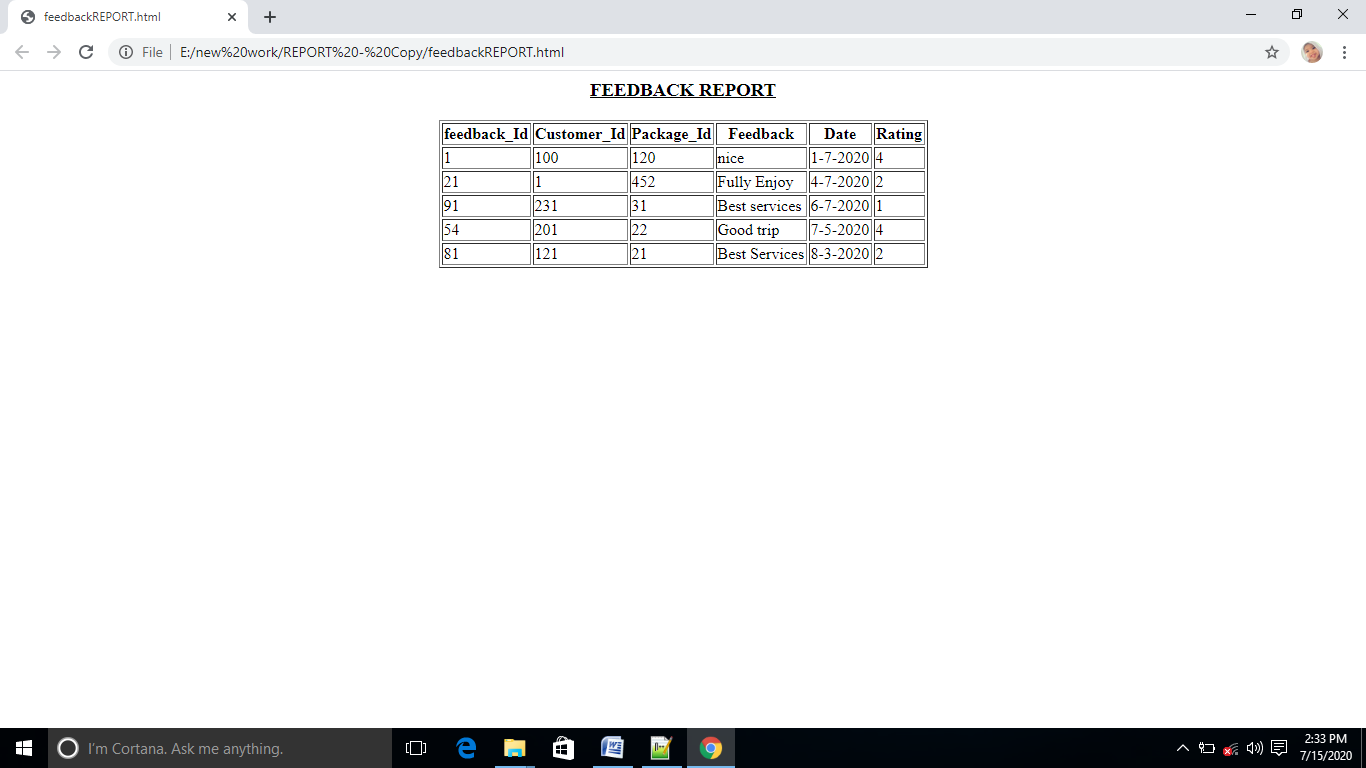


Figure Feedback report

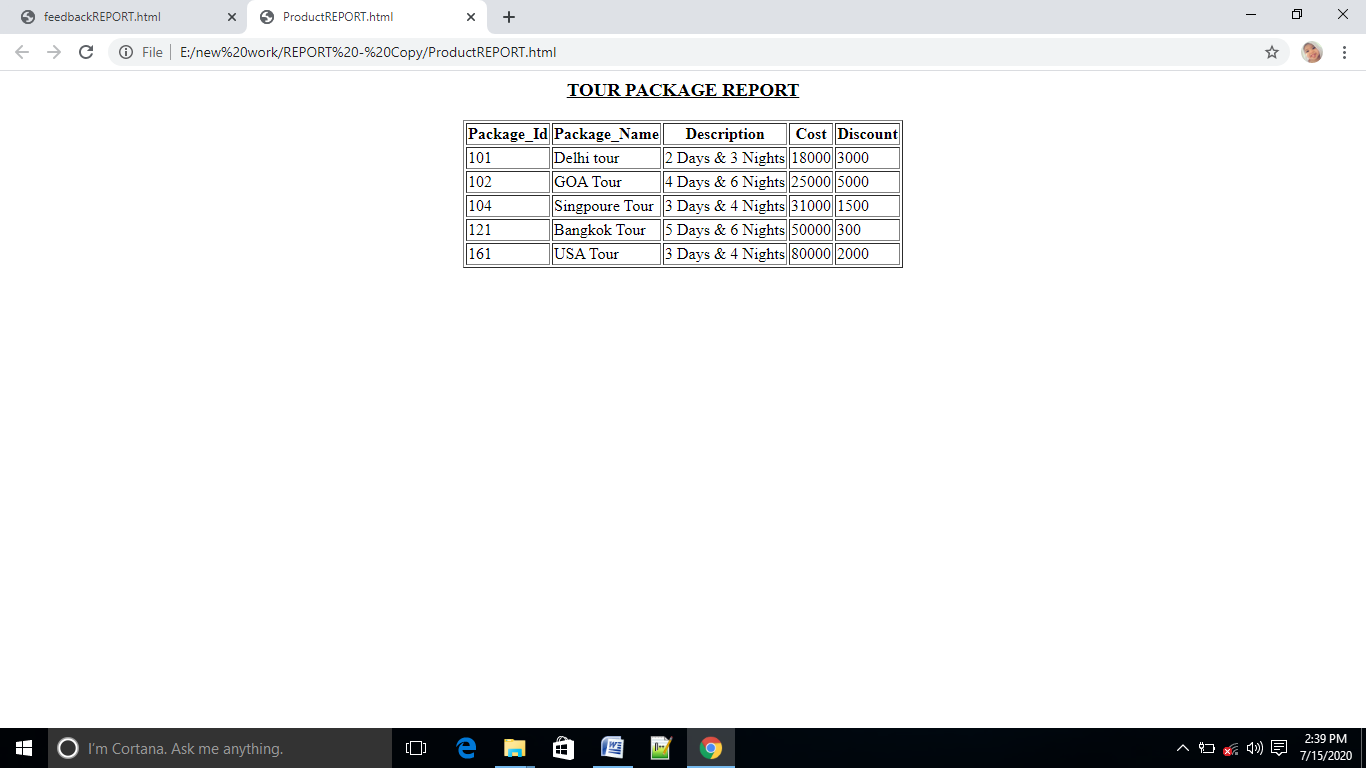


Figure Tour and Package report 7.IMPLEMENTATION

**7.1 STANDARDIZATION OF THE CODING/CODE EFFICIENCY**

In my project I have used various controls like command buttons, hover buttons and text box and similar other elements from various classes. Standardization of coding is a very good technique which should be followed while coding the project. Standardization of coding has various advantages over non-standard code such as:

* Standard code can be easily understood
* It is easy to debug
* It is easy to modify
* It is easy to upgrade

In this, project, much care has been given to in developing the standard program code. For example, each java class name start with uppercase letter and each method starts with a lowercase letter, but other words of method except first one starts with uppercase letter. All JSP files are stored in a folder named jsp, all bean classes are stored in a folder named beans with in src package. And all the contents of project code files have been stored according to a proper directory structure***.***

Code of the software is said to be efficient, if the complexity of all types is minimum. In the code of the developed software, I have tried to minimize the space and time complexity, so the code is efficient.

The complexities can be minimized in three ranges. They are called as organized simplicity, disorganized complexity and organized complexity.

Organized complexity is methodologically undeveloped in the sense that neither analytical nor statistical methods are adequate for dealing with systems that fit into it.

**Computational complexity:** Computational complexity is a characterization of time and space requirements for solving a problem by a particular algorithm. Either of these requirements is usually expressed in terms of a single that represent the size of the problem.

Although computational complexity has been predominantly studied in terms of the time it takes to perform a computation, the amount of computer memory is required frequently just as important. This requirement is usually called the space requirement. It is expressed in the terms of a space complexity function, analogous to the time complexity function.

Since the modular approach is used to design the software. This approach uses the object oriented design methods. Hence the code is optimized due to the above reason. Because of object oriented programming, the features like modularity and reusability can be achieved in the software.

Reusability means – the programmer modifies the program’s functionality by replacing the old elements or objects with new objects or by simply plugging new objects into the application. General instruction requires no modification because specific implementation details reside within the object.

**Optimization of coding:** Optimization of code is very important to produce the better quality of software. Without optimization it is very difficult to debug the syntactical and logical errors present in the code. If the code will be optimized then it becomes trouble-free to find out the errors is stipulated time.

Every step has been taken to optimize the coding. The efforts has been made to modularized the whole working of the software by which it will be easy to locate the errors is time saving manner. Any modification in the requirement of users can also be implemented if the code of the software will be optimized.

For the purpose of code optimization the existing code has been re used, rather than doing all the coding from scratch. For example, all the bean classes and java classes are kept in source package and these are imported on the JSP pages whenever required.

7.2 Exception Handling

When an error occurs, or exception as we call it, Python will normally stop and generate an error message.

These exceptions can be handled using the try statement:

Example

The try block will generate an exception, because x is not defined:

try:  
  print(x)  
except:  
  print("An exception occurred")

Many Exceptions

You can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error

Else

You can use the else keyword to define a block of code to be executed if no errors were raised:

Finally

The finally block, if specified, will be executed regardless if the try block raises an error or not.

**Example**

try:  
  print(x)  
except:  
  print("Something went wrong")  
finally:  
  print("The 'try except' is finished")

7.3 PARAMETERS CALLING/PASSING

Passing parameters from one web page to another is a very common task in Web development. There are many situations in which we need to pass data from one Web page to another. There are many techniques available for this purpose. The most common techniques used for this purpose are URL rewriting, hidden form filled, session management, setAttribute() method of HTTP request etc. In this project “Tour and Travels”, parameter passing has been done using usingurl re-writing, session and setting attribute in request. However, every method of parameter passing its own limitations and scopes. Using url re-writing has disadvantage that it the passing parameters would be shown in the address bar, so it can’t be used for passing sensitive data. Setting request attribute for passing parameter has limitation that parameter will only be available to the requested page and not elsewhere. HttpSession is used when passing parameters specific to the client Session has number of advantages over the url re-writing and also some disadvantages. The comparison of these is described below-

* Session works on server side, while url re-writing work on client side.
* The information or data stored in query string in url re-writing is visible to everyone. But in Session it is hidden and can’t be viewed easily.
* Query String in url re-writing can store only a piece of information but in Session we can store the more and more data.
* The Query String speed never falls as the load increase because it stores a piece of information. But on the other hand Session increase congestion as the loads increase.

7.4 VALIDATION CHECKS

For every system, validations play a very important role while accepting the inputs from the users. This is because, the data being input by the user is further used to keep track of the various activities and their accounts.

In the **Vaccination Scheduler** the user inputs are validated because that data is then further used for generation of reports, for verification etc.

In this system two languages are used for validation checks.

**Java Script** is used for validating the user inputs for client site verification. Because user have to insert appropriate data in the data fields. Such as if a user inputs the email id in data field they have to input appropriate email id or if a user inputs name then they cant use number in name.

**JSP** is used for validating the user inputs from database because a proper profile id cannot be inserted if the user\_profile is already inserted.

The main validations that are done in **Vaccination Scheduler** are as follows:

1. All the screens have a similar look and feel. They all have the almost same color combinations in its background. This provides a better user interface to the users.
2. Whenever a page needs to be refreshed, only a required portion needs to be updated, instead of the whole page. This helps in fast refreshing of the page.
3. Whenever a user logs on to the **Vaccination Scheduler** through the Login, his/her rights and privileges are checked and then he/she is allowed to work under the permitted rights only.
4. The data entered by the users is validated before the saving or retrieval of the record.
5. The user, except that administrator, is not allowed to message directly to the other user.
6. Password field contains at least 6 character long data, which stored in the database in the encrypted form.
7. If any data entered by users are wrong then it will shows an error message without proceeding any process.
8. Numeric field only contains numbers like age, mobile no, enrollment no, session, experience, day, month, year field. If any user entered alphabets then it will display an error message to acknowledge the user that he/she is given a wrong entry.
9. After logout, you will have to re-sign-in for using your account.
10. Alphabetical field only contains only alphabets like first name, last name. If any user entered number then it will display an error message to acknowledge the user that he/she is given a wrong entry.
11. User can only fill appropriate email\_id in email\_id field. If any user entered inappropriate form of email\_id then it will display an error message to acknowledge the user that he/she is given a wrong entry.
12. User should have to enter differ profile\_id from existing user.

Thus, we have tried to make this system very secured and reliable by putting a number of validation checks in it. The future versions of this software are supposed to have more extended validations checks based on varied client needs.

**Blank Field validation**

Most of the fields must not be left blank. If user left blank a mandatory field then an alert message will be displayed and user will be prompted to fill it. Those fields are mandatory to fill before processing the data.

**Date Validation**

The validation on date data type has been specified to be of the format DD/MM/YYYY. Any other format is unacceptable.

**Number field validation:**

The field specified with number as then their data- type will not accept character. For example, mobile number always must be numeric.

**Character field validation:**

The field specified with character as then their data- type will not accept numeric. For example- first name always must be character.

**E-mail field validation**

When user will input his/her e-mail either to login or register, it will be checked that if it is a valid e-mail or not. If not a valid e-mail then, application will show a alert message and prompt to input again.

**User Authentication**

When Employee/Customer/Admin logs on to the system to access data from tables and database, the Id, password and proper login type needs to be checked.

**Password change Validation**

Only authorized users are allowed to change the password and the process requires asking the old password before changing it to the new one.

8. Testing

The development of software systems involves a service of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as later design and development stages. Because of human inability to perform and communicate with perfection, software companies are accompanied by a quality assurance activity.

Testing is to determine errors in a software code. It is crucial element of software quality assurance and represents the ultimate review of specifications, design and coding. The increasing visibility of a software as a system element and the attendant “costs” associated with a software failure are motivating forces for well-planned through testing. Usually software development organizations expend between 30 to 40 percent of total project effort on testing. Our goal is to design a series of test cases that have a high likelihood of finding errors.

To test the software, there are so many testing techniques which provide systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

If testing is conducted successfully, it will uncover errors in the software. As the secondary benefits, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appear to have been met.

8.1 TESTING STRATEGIES

Designing effective test cases is important but so is the strategy we use to execute them. A strategy for software test case design methods that result in the successful construction of software. The strategy provides a road map that describes the steps to be conducted as a part of testing.

There are a number of testing strategies, which have the following generic characteristics: -

* Testing begins at the component level and works “outward” toward the integration of the entire computer–based system.
* Different testing techniques are approximate at different points in time.
* Testing is conducted by the developer of the software and (for large projects) an independent test group.
* Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

System engineering

Requirements

System testing

Validation testing

Integration testing

Initially, system engineering defines the role of software and leads to software requirement analysis where the information domain, function, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to the design and finally to coding. There are a number of testing strategies, which are given: -

**Unit testing**

In the unit testing interfaces, local data structures, boundary conditions, independent paths, error-handling paths are tested. Test cases should be design to uncover errors due to erroneous computations, incorrect comparisons, or improper control flow. For this purpose basis path and loop testing is done. After source level code has been developed, reviewed and verified for correspondence to component level design, unit test case design begins. In unit test application ‘drivers’ are developed which are programs, accept test case data, passes such data to the component to be tested and prints relevant results. ‘Stubs’ are also developed which serve to replace modules, that are subordinate the component to be tested.

Six Rules of Unit Testing:

1. Write the test first
2. Never write a test that succeeds the first time
3. Start with the null case, or something that doesn't work
4. Don't be afraid of doing something trivial to make the test work
5. Loose coupling and testability go hand in hand
6. Use mock objects

**Limitations of unit testing**

Testing, in general, cannot be expected to catch every error in the program. The same is true for unit testing. By definition, it only tests the functionality of the units themselves. Therefore, it may not catch integration errors, performance problems, or other system-wide issues. Unit testing is more effective if it is used in conjunction with other software testing activities.

Like all forms of software testing, unit tests can only show the presence of errors; it cannot show the absence of errors.

Software testing is a combinatorial problem. For example, every Boolean decision statement requires at least two tests: one with an outcome of "true" and one with an outcome of "false". As a result, for every line of code written, programmers often need 3 to 5 lines of test code. Therefore, it is unrealistic to test all possible input combinations for any non-trivial piece of software without an automated characterization test generation tool or many of the tools listed in List of unit testing frameworks.

To obtain the intended benefits from unit testing, a rigorous sense of discipline is needed throughout the software development process. It is essential to keep careful records, not only of the tests that have been performed, but also of all changes that have been made to the source code of this or any other unit in the software. Use of a version control system is essential. If a later version of the unit fails a particular test that it had previously passed, the version-control software can provide a list of the source code changes (if any) that have been applied to the unit since that time.

It is also essential to implement a sustainable process for ensuring that test case failures are reviewed daily and addressed immediately. If such a process is not implemented and ingrained into the team's workflow, the application will evolve out of sync with the unit test suite—- increasing false positives and reducing the effectiveness of the test suite.

**Integration testing**

Integration testing is systematic technique for constructing the program structure while at the same time conducting the tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. There are two types of integration – Bottom up integration and Top down integration. Regression and smoke testing are done in integration testing strategy.

**Validation testing**

Next step is the validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. At the culmination of integration testing, software is completely assembled as a package, interfacing errors has been uncovered and corrected, and a final series of software tests i.e. validation testing begins. Validation succeeds when software functions in a manner that can be reasonably expected by the customer.

Software validations can be achieved through a series of black-box tests that demonstrate conformity with requirements. Configuration review, alpha testing and beta testing techniques are conducted to enable the customer to validate all requirements. Acceptance tests are conducted by the end-users over a period of weeks or months.

**System testing**

Finally, we arrive at system testing where the software and other system elements are tested as a whole. System testing verifies that all elements mesh properly and that overall system function / performance is achieved. Ultimately software is incorporated with other system elements and a series of system integration and validation tests are conducted.

9. Bibliography

* HTML/DHTML Complete Reference (TMH)
* JavaScript Complete Reference (TMH)
* JavaScript (O’ Reilly)
* JavaScript (Wrox)
* JavaScript and DHTML cookbook (O’ Reilly)
* Learning Python (O’Reilly)
* Learning Python by Mark Lutz
* Python Essential Referenced by David Beazley
* Programming in Python 3 (Mark Summerfield)
* Think Python (O’Reilly)
* Python for everybody (CHARLES SEVERANCE)
* Python for Beginners 2017