**A Project Report**

**on**

***Tourism Management System***

Submitted in the partial fulfillment of the requirements

Of the award of the degree of Bachelor in Computer Applications.

****

**Guide- Submitted By-**

``````` gghjjjjjjjjjjjjj

Roll- 1661330061

**Study Center:**

SoftTech Software Pvt. Ltd, Haldwani (Nainital)

**DECLARATION**

I hereby declare that the project entitled “**Tourism Management System**” submitted for the BCA degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

|  |  |  |
| --- | --- | --- |
| **NAME** | **UNIVERSITY ROLL** | **CLASS** |
|  | 1661330061 | BCA 6th semester |

**Place:** Haldwani

**Date:**12.06.2019

**CERTIFICATE**

This is to certify that project entitled “Tourism Management System” in partial fulfillment for the award of the degree of **Bachelor in Computer Applications** through **Amrapali Institute of Management and Computer Applications** done by **”name ur ”**, Roll No. **1661330061** is an authentic work carried by him at **SoftTech Software Pvt Ltd.**under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

**Date**:12.06.2019

**Place**: Haldwani

**Name of Guide**: Gaurav Gahtoti

**Signature of Guide**:

SoftTech Software Pvt Ltd (Haldwani)

**ACKNOWLEDGEMENT**

I express my gratitude to my mentor and guide Mr. Gaurav Gahtori for helping in the completion of this project work.

I am deeply obliged to my teachers at Amrapali for their teachings and suggestion by which this project has been made successful.

I would like to thanks my entire team-mates for their co-ordination, encouragement, suggestion and sometime criticism which surely helped in the completion of our work.

I cannot forget my friends for providing me ideas and encouragement throughout this project work. Big thanks to them.

Finally I would like to thank everyone whom I forget to mention for making this project work a success.

**INDEX**

Declaration

Certificate

Acknowledgement

Project Summary

Abstract

1. Introduction
   1. Tourism Management System
   2. Purpose
   3. Scope
   4. Objective
   5. Advantages
   6. Definition, Acronyms, Abbreviations
   7. References
   8. Overview
2. Feasibility Study

Feasibilty Analysis

* 1. Economic Feasibility
  2. Technical Feasibility
  3. Market Feasibility

1. Requirement Analysis
   1. Hardware and Software Requirements
   2. Proposed System
   3. Need for Computerization
   4. Existing System
   5. Input & Output
   6. Software System Attributes
2. Project Category
3. Tools and Platform
   1. Platform
   2. Database
   3. Development Tools
   4. Architecture
4. System Design
   1. Flowchart
   2. DFD
      1. Level-0 DFD
      2. Level-1 DFD
   3. Entity Relationship Diagram
      1. Logical Entity Relationship Diagram
      2. Physical Entity Relationship Diagram
   4. Use Case Diagram
5. I/O Form Design
   1. Input Design (Introduction)
   2. Elements of Input Data
   3. Input Data
   4. Source Documents
   5. Input design guidelines
   6. Objectives of Input design
6. Database Design
7. Snapshots of Product
8. Testing
   1. Definition
   2. Testing Methodologies
   3. Types of Testing
   4. Test Cases
9. Conclusion

**PROJECT SUMMARY**

Title :Consultancy & Recruitment

Definition : To automate the process of hiring.

Developed At : Amrapali Institute of Management & Computer Applications

Team Size : 4

Team Members : Pranay Singh Negi

: Pawan Singh Babiyari

: Naresh Bhatt

: Naveen Chandra Satyawali

**DEVELOPMENT TOOLS**

* Operating system : Windows 10
* Front end Languages : HTML, CSS, JavaScript
* Back end Language : Php & MySQL
* Documentation Designing : Microsoft office 2013 and MS Word.
* Database Designing : phpmyadmin

**ABSTRACT**

The main objective of this Tourism Management project is to make the holiday easy and comfortable for the users right from finding perfect destination for theirselves till the bookings of the tickets. In the project, we have two modules: Admin and User.

Admin in this ‘Tourism management System’ project will log in with the admin username and password and admin has the authority to add various holiday packages, the admin will also provide the holiday package details, prices and various other important details required to know by the user while searching the suitable holiday destination.

Finally, the users will register and log in and find the appropriate tourist destination for theirselves. The user can book a package for self and the system will generate a booking ticket as to confirm the booking for the user. The user can also enquire about various packages, or any other queries related to holiday by a contact portal available in the system.

1. INTRODUCTION
   1. **Tourism Management System**

Online Tourism Management System is an integrated web-based software developed to make tour easy for the travellers who want to get best experience on their holidays. The main aim of this project is to promote regional culture and to connect many people to it through tourism.

This online system make tourism management easy by handling customers’ requests and providing services for the customers located at different parts of globe. This system bring all the convenient facilities in one place where customer can look for their favourite destination, enquire about them, and book the perfect package in a hassle free environment.

Different modules have been incorporated in this project to handle different parts and sectors of tourism management field. A customer can book the package anytime and will get a confirmation receipt when the booking get confirmed. A user can also cancel or print the receipt of booking anytime through the profile page.

This system is very user friendly and is designed is such a way that any type of user can easily access it without any difficulties.

Since this is online website thus, it can be accessed through any device, through any web browser and available 24 X7. What is needed is just an **INTERNET**!!

* 1. **Purpose**

The purpose of this project is to automate the task of tourism booking so to avoid using much manpower, money, reduce time etc. This system can do such task more efficiently than the manual system, thus proves beneficiary for its users. Customers register theirselves and then book a desirable holiday destination. A confirmation receipt is generated which can be downloaded and to be brought along at the time of departure.

* 1. **Scope**

The product will work by a Tour company Admin and Customer. This project named as Tourism Management, will provide Tour related facilities to its users. Users book their desired tourist destination which they can also cancel if they do not wish to have later. The system will help all the people who wish to travel in finding a perfect destination for theirselves.

* System aims to improve the current process of tourism management.
* Allows its users to view all the available destination.
* System also maintains the record of previous tours and cancelled tours.
* It also provide an enquiry system where user can get all the solution to their queries.
  1. **Objective**

The Business objective of the system is to save the time of peoples looking for places to travel. Having Online Tourism Management System will reduce the stress of collecting details best location. Users will be able to comfortably plan and book perfect holiday with just a click.

* Making the Tour booking processes simpler.
* Improve Customer (traveler) satisfaction.
* To provide the receipt for the confirmed booking.
* Reduce the un-necessary paperwork, manpower, time.
  1. **Advantages**

Before implementing such a system, its benefit versus its cost needs to be considered. The advantages of having Online Tourism Management System are listed below:

1. A user can make a booking anytime from anywhere easily.
2. Reduces paper work and time used as compared to manual process.
3. Ability for travelers to track authorization and reimbursement request status through the system rather than via phone calls or campus mail.
4. A user will be able to take bookings around the clock, 7 days a week.
5. Reduces detrimental environmental impacts

There are some other advantages in terms of safety, security which makes it easier for users. Also, economical and efficient space utilization and environmentally friendly system.

* 1. **Definition, Acronyms And Abbreviations**

**Personal Details**: Details of users such as userid, username, phone number, e-mail address etc.

**Contact Details**: Details of contact associated with the users.

**SRS**: System Requirement Specification

**MY SQL**: the database

* 1. **References**
* Software Requirement Specifications from Internet.
* [www.youtube.com](file:///G:\www.youtube.com)
* [www.stackoverflow.com](file:///G:\www.stackoverflow.com)
* [www.tutoruialspoint.com](http://www.tutoruialspoint.com)
* [www.geeksforgeeks.com](http://www.tutoruialspoint.com/)
  1. **Overview**

**User Types:**

* User
* Admin

There are two kind of users who will have the un-interrupted access to the system data. The whole working starts by registration of the user by recording some personal details. The **‘Admin’** will be the one who control all of the website. An **‘Admin’** will add tour packages and related details. The **‘user’** will login and will able to explore all the tour packages filled in by Admin. The user will then book or enquire about the tour packages. If User has booked a package, a ticket for the same will be generated which would be printed-out by the user and should be brought along by the user on the departure date. If user has to cancel the booking, which can be done by visiting profile menu where user can not only cancel the bookings but also check the upcoming, past or cancelled bookings.

The profile menu is also provided where both **User & Admin** can check and update their details.

1. Feasibility Study

**Feasibility Analysis**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

* 1. **Economic Feasibility**

The proposed or new system will add a lot of economic value to the Tourism Industry and also to the Travelers. Based on the given information, there are no exact figures on the amount of profit generated by adopting the system, but certainly it will do. However, there are information indicating the faults of the current manual system, which discourage some users to proceed with the current system. Paper work, time and more manpower are three certain factors which discourage the use of current manual system. The new system will alleviate the faults of the current system hence retaining the old and attracting user base. The new system will turn the old users into repeat or loyal customers who will always be utilizing the product despite of economic conditions. The new system will also attract new customers who will help in maximizing the profit. The system don’t require additional hardware or supporting software to be needed which make it a profitable deal.

* 1. **Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

• Does the necessary technology exist to do what is suggested?

• Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?

• Can the system be upgraded if developed?

• Are there technical guarantees of accuracy, reliability, ease of access and data security?

The current system collects processes and stores many candidates data utilized in the day-to-day operations of the working of the system. However, the current system is associated to various faults.The new system requires a good infrastructure both hardware and software to enable it overcome these challenges successfully. The software has to be able to process the big data due to the large number of candidates while the hardware has to be able to store the large data. The Users needs to be a little technical so as to operate the IT infrastructure utilized by the system. Good IT infrastructure and technically skilled users will ensure that product will work without any problem.

* 1. **Market Feasibility**

The market issue usually raised during the feasibility stage of the investigation includes the following:

• Does the technology has outdated?

•Does the market have already developed the better version of the software with same objectives?

• Is there enough market space required by software to sustain?

A lot of companies which now a days are registering in the market, most of them are new startups. These requires the new talent as fast as possible, so that they do not lag in the race of capturing the market to maximize the profit. Same case also exist with the well settled companies. This increases the demand of such system which can meet the market challenges. Tourism has a great demand and thus use of this system will be frequently and even more frequently. Thus causing the future safe demand of this system for ever.

This system will help the Company recovers the funds utilized in adopting the new system, maintaining it. That adoption cost will be much smaller, as this product comes under everyone pocket.

1. REQUIREMENT ANALYSIS

In this, we are going to discuss the requirements, design of the system and analyze them using some appropriate software models.

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In [software](https://searchmicroservices.techtarget.com/definition/software) engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mould user expectations to fit the requirements. Requirements analysis is a team effort that demands a combination of hardware, software and human factors, engineering expertise as well as skills in dealing with people.

* 1. **Hardware And Software Requirements**

Hardware Specifications (Minimum):-

* Processor: X86 Compatible Processor with 1.7GHz Clock Speed
* RAM: 512MB
* Hard Disk: 20GB
* Monitor: VGA/SVGA
* Keyboard: 104 Keys
* Mouse: 2buttons/ 3buttons

Software Requirements (Minimum):-

* Operating System: Cross Platform.
* Internet connection
* Any web Browser which is updated to its latest version.
  1. **Proposed System**

**Operational Requirements:-**

* The system will present the various data in front of the user by which the user can take action upon it.
* The system will update information to the main databases.
* The system shall require individual (Admin and User) log-in credentials for each user in-order to access their accounts.
* The system shall be accessible in any Platform through any web-browsers.
* The system shall be compatible with low specification desktop systems.

**Performance Requirements:-**

* The system shall be available 24/7
* When data / information is requested it shall be presented on screen within 3-5 seconds
* The system should be able to handle any kind of exception.
* System shall be able to accommodate simultaneous users at a time

**Security Requirements:-**

* The system shall ensure all data is protected from unauthorized access
* The system shall comply with all PCI regulations
* User password must abide with DIT-A Password policy
  1. **Need for Computerization**

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

* Minimizing the manual records kept at different locations.
* There will be more data integrity.
* Facilitating desired information display, very quickly, by retrieving inform from users.
* Facilitating various statistical information which helps in decision-making?
* To reduce manual efforts in activities that involved repetitive work.
* Updating and deletion of such a huge amount of data will become easier.
  1. **Existing System**

The present systems are inadequate in providing information and advices to the agencies and customers about tour plans. Often companies are compelled to rely on local information sources and count on their own experience regarding time and cost. Through phone call they have to get information. There are some problems which are existing in the present system those are given below concisely:

* Local Tourist spots are neglected by various tourism websites.
* Transparency is not maintained.
* It take much time to plan a holiday.
  1. **Input & Output**

There are 2 kind of users who are interacting with the system:

1. **Admin**

Does Login and gain access to:-

* Register Users with their details
* Add new holiday package
* View Users Enquires
* View Bookings
* Edit / Delete Package
* Cancel Bookings done
* Update Profile

1. **User**

Does Login and gain access to:-

* View Holiday Packages
* Enquire related to holidays
* Book holiday Packages
* View Past Bookings
* Cancel Upcoming Bookings
* Update Profile
* View Bookings Tickets
  1. **Software System Attributes**

There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified. The following items provide a partial list of examples. These are also known as non-functional requirements or quality attributes. These are characteristics the system must possess, but that pervade (or cross- cut) the design. These requirements have to be testable just like the functional requirements. It’s easy to start philosophizing here, but keep it specific

* **Reliability**

It means the extent to which program performs with required precision. The website developed should be extremely reliable and secure so that information about any questions etc. is not leaked. The system shall not be down more than 2 times in a year.

* **Availability**

The software will be available only to authorized users like user to view their details, admin to add an update/delete user details and book details. Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevailing availability problems.

* **Security**

The security requirements deal with the primarily security. The software should be handled only by the administrator and authorized users. Only the administrator has right to assign permissions like creating new accounts and generating password.

Specific requirements in this area could include the need to:

* Utilize certain cryptographic techniques
* Assign certain functions to different modules
* **Portability**

The software is a web based application and is built in HTML & Php. So it is platform independent and is independent of OS. The application will be easily portable on any system.

* **Maintainability**

The application is to be designed so that it is easily maintained. Also it should allow incorporating new requirements in any module of system. Backups for database are available.

1. PROJECT CATEGORY

Our Project Belongs To The Three Tier Architecture Category.

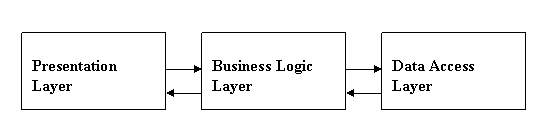
Presentation Layer:-

This layer will mainly consist of user interface of the system i.e. the visible part of a system. Which in our project is made using .Net Tools and requires a Window OS to work upon.

Business Logic Layer:-

This layer acts as a mediator between Presentation layer and Data Access layer. This layer will have the business logic implementations and it transfers the data between the two layers.

Data Access Layer:-

This layer deals with the database and database related coding. We writes queries in this layer.

1. TOOLS / PLATFORM
   1. **Platform:**

Any Operating System

(A basic platform for running applications)

* 1. **Database:**

**My SQL**

My SQL is a database management system that combines the relational Database with a graphical user interface and software-development tools.

* 1. **Development Tools:**

The tools used to develop include:

HTML5, JavaScript, Php, CSS

- My Sql

* 1. **Architecture**
     + If you have worked with other database systems on desktop computers, you might have seen the term database used to refer to only those files in which you store data.
     + If you have worked with other database systems on desktop computers, you might have seen the term database used to refer to only those files in which you store data.
     + But, in Access, a desktop database (.accdb) also includes all the major objects related to the stored data, including objects you define to automate the use of your data.

**An Overview of HTML**

**A simple HTML Program**

In HTML, FORM tags can be used to interact with users. Things that can be added to a form are the likes of text boxes, radio buttons, check boxes, drop down lists, text areas, and submit buttons. A basic HTML form with a textbox and a Submit button looks like this:

<html>

<head>

<title>A BASIC HTML FORM</title>

</head>  
<body>

<form name ="form1" **method** ="POST" **action** = "">

<input type = "text" value ="username">

<input type = "Submit" Name = "Submit1" value = "Login">

</form>

</body>

</html>

The attributes like **METHOD**, **ACTION** and **SUBMIT** used in the form are important to study.

* The **Method** attribute is used to tell the browser how the form information should be sent. The two most popular methods you can use are GET and POST. But our METHOD is blank.
* As long as SUBMIT has an **ACTION** set, then your data will get sent somewhere.
* The HTML **Submit** button is used to submit form data to the script mentioned in the ACTION attribute.

The work is then saved as “**basicForm.php”**.

Start your server, and make sure the form loads ok in your browser. You should be able to see a text box and a Submit button.

**An Overview of Php**

**A Simple PHP Program**

Beginning with a very simple Php program. Create a file (named random.php):

<html>

<head><title>Random number</title>

</head>

<body>

<p>I have chosen <?php echo random(1, 100); ?>.</p>

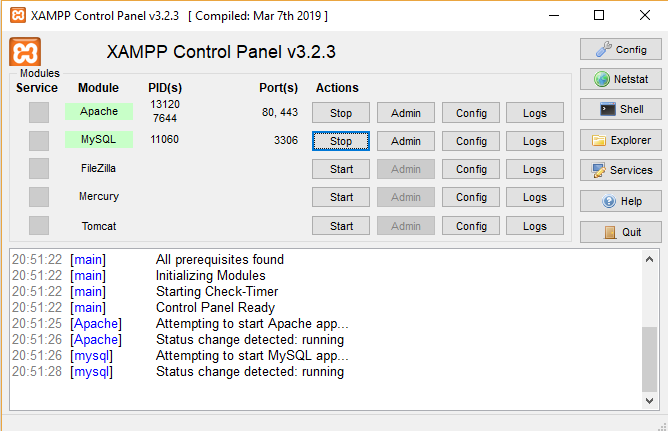
</body>

</html>

We can install it on the Web server; if the Web server supports PHP, then when we load it into your Web browser we'll see output.

**Saving PHP File**

Whenever you create a new PHP page, you need to save it in your htdocs directory of Xampp folder located on C: drive. You can see where this is by clicking its item on the menu:



**The echo statement**

PHP programs consist of a sequence of **statements**, which typically end with a semicolon. In processing PHP code, the server processes the first statement; then it goes to the next statement; then the next; then the next; until it reaches the end of the PHP. (You can see why a semicolon is appropriate for ending each individual statement.)

The PHP portion of the above Web page contains just a single statement: **echo**random(1, 100); This is an example of an **echo** statement. When the Web server reaches an **echo** statement, it knows that it will send some information to the Web browser. Whatever comes between the word **echo** and the semicolon is what it should send; in this case, it sees that it should whatever number random generates.

**Variables and Functions**

A variable is just a storage area. You put things into your storage areas (variables) so that you can use and manipulate them in your programmers. An **assignment statement** — the second type of statement that we've seen (the first was the **echo** statement). Here is a simple example:

$pi=3.1415;

An assignment statement begins with the variable name, followed by an equals sign, followed by the value to give it;and it ends with a semicolon. In this case, the variable I am creating is named $pi and I am assigning the name to refer to the value 3.1415.

**The $\_GET Function**

The built-in $\_GET function is used to collect values from a form sent with method="get".

Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

**Example:**

<form action="welcome.php" method="get">  
Name: <input type="text" name="fame" />  
Age: <input type="text" name="age" />  
<input type="submit" />  
</form>

**The $\_POST Method:**

The built-in $\_POST function is used to collect values from a form send with method="post".

Information sent from a **POST** method is invisible to others and has no limits on the amount of information to send.

**Example:**

<form action="welcome.php" method="post">  
Name: <input type="text" name="fname" />  
Age: <input type="text" name="age" />  
<input type="submit" />  
</form>

**My SQL Functions**

PHP provides access to MySQL databases via a number of functions. We'll find six of them particularly useful. They are listed below in the order they will typically be used.

1. **mysqli\_connect( ) Function:**

The mysqli\_connect ( ) function opens a new connection to the MySQL server.

**Syntax:**

mysqli\_connect (host, username, password, dbname, port, socket);

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *host* | Optional. Specifies a host name or an IP address |
| *username* | Optional. Specifies the MySQL username |
| *password* | Optional. Specifies the MySQL password |
| *dbname* | Optional. Specifies the default database to be used |
| *port* | Optional. Specifies the port number to attempt to connect to the MySQL server |
| *socket* | Optional. Specifies the sockets or named pipe to be used |

1. **mysqli\_query( ) Function:**

The mysqli\_query ( ) function performs a query against the database.

**Syntax:**

mysqli\_query (connection, query);

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *connection* | **Required**. Specifies the MySQL connection in use |
| *query* | **Required**. Specifies the query string |

1. **mysqli\_error( ) Function:**

The mysqli\_error ( ) function returns the last error description for the most recent function call, if any.

**Syntax:**

mysqli\_error (connection);

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *connection* | **Required**. Specifies the MySQL connection to use |

1. **mysqli\_num\_rows( ) Function:**

The mysqli\_num\_rows ( ) function returns the number of rows in a result set.

**Syntax:**

mysqli\_error (result);

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *Result* | **Required**. Specifies a result set identifier returned by  mysqli\_query();  mysqli\_store\_result() or  mysqli\_use\_result() |

**Connect to MySQL database from PHP**

MySQL and PHP are two technologies used by many web developers. Connecting to MySQL from PHP is easy and when we consider that both PHP and MySQL are free technologies, the pair sounds like winning choice. This article shows how to connect to MySQL with PHP.

The PHP script below simply connects to a MySQL database and then immediately closes the connection. Of course you will want to do more than that – reading, updating, inserting and deleting data, but once you are connected to the database, these tasks are easy.

<? php

$Conn = mysqli\_connect (hostname, username, password, dbname)

Or die ("Connecting to MySQL failed");

mysqli\_close ($Conn);

?>

Now that you have connected to the MySQL server from your PHP script, how do you select which database to work with (there can be more than one database on the same MySQL server)? Here is how to select a MySQL database to work with, from PHP:

<? php

$Conn = mysql\_connect ($hostname, $username, $password)

Or die ("Connecting to MySQL failed");

mysqli\_select\_db ($database, $conn)

Or die ("Selecting MySQL database failed");

mysqli\_close ($conn);

?>

Now that we have selected a MySQL database, we can retrieve some data from it and send it back to the browser from our PHP script:

<?php

$Conn = mysqli\_connect ($hostname, $username, $password)

Or die ("Connecting to MySQL failed");

mysqli\_select\_db ($database, $Conn)

Or die ("Selecting MySQL database failed");

$sSQL = “SELECT First Name, Last Name FROM Users";

$result = mysqli\_squery ($sSQL, $Conn);

While ($row = mysqli\_fetch\_object ($result)) {

echo $row->First Name . " " . $row->Last Name. "";

}

mysqli\_close($conn);

?>

In the PHP script above we are connecting to MySQL database, selecting data with mysqli\_query and printing it to the browser.

**Approaches Used:**

Top- down and bottom-up approach are strategies of information processing and knowledge ordering.

A **top-down** approach is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in yet greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements.

A **bottom-up** approach is essentially piecing together systems to give rise to grander systems, thus making the original systems sub-systems of the emergent system. In a bottom-up approach the individual base elements of the system are first specified in great detail. These elements are then linked together to form larger subsystems, which then in turn are linked, sometimes in many levels, until a complete top-level system is formed.

**Online Tour&Travel Process**has used **top-down** approach. In this system, firstly understand the problem. The major step in the development process is to realize exactly what the problem to be solved is really about. These first thoughts give you valuable information about the purpose of the application. Another valuable source of information is the end users that are going to use the application. It is always a good idea to involve the end users at an early stage of the development process to make sure that your application solves the correct problem. Now system is divided into subsystems like admin system, customer system. Create a database. In this database, create tables for all subsystems like registration, ordering, order check and proceed etc.. Specifying each and every field of all these tables in detail. Specify the primary key and foreign keys of each table in the database. Now specify the relationships between these tables in the database. Specify which field of which table is related with which field of another table and how. In this way, link each table of the database with each-other. Thus, the development of Online Restaurant Process follows top-down approach.

By identifying how the application is going to be used and which routines the users will perform with the system, you get a good idea of the data your application is going to handle. Designing the tables will often be the most difficult part of the entire application development process. Getting your table design right is crucial as the rest of the application is built on top of the tables.

It is also important that you identify the forms and reports that are needed. This is the only way you can design the underlying system to provide the information necessary in them.

The most important results of this step are a clear picture of...

1. the problem the system is going to solve
2. the functionality of the system, that is, how will the application solve the problem
3. who is going to use the system

Other results of this step are first ideas of...

* which tables are necessary (and relations between them)
* which forms and reports the Customer need
* the structure of the menus that gives access to the entire application

When you have completed this step, test your results with the users/customers. Create a requirements specification together with your customer. The requirements specification can be made as a description of criteria which must be fulfilled.

**Create tables to hold our data**

Our tables and the relations between them make up the foundation of our application. A bad table design affects the whole application. This is why it is important to analyze our data and break it into tables before you create any other application objects

Defining relations between our tables are very important. You should make sure that our tables adhere to the first three normal forms. This brings many benefits, such as an increase in consistency and maintainability, a reduction in data redundancy and better search performance from smaller tables (with narrower rows).

1. SYSTEM DESIGN

**Methodology Adopted**

We prefer iterative waterfall model for our project as it includes all the activities that are

* Feasibility study
* Requirement analysis and specifications
* Design
* Coding and unit testing
* Maintenance

In iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases.

The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phase. But there is no feedback path to the stage feasibility study, because once a project has been taken, does not give up the project easily. It is good to detect errors in the same phase in which they are committed. It reduces the efforts and time required to correct the errors.

That is why we prefer this model for our projectbecause it includes many good points which make our project more effective than ever sinceand it reduces our time which is really veryimportant point.This model is very easy to understand and use.That is why it is one of the most widely usedsoftware development models.

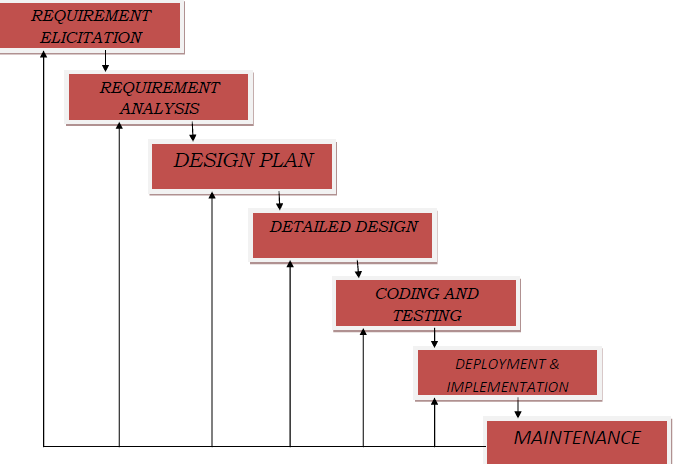
**The SDLC Waterfall:-**

Small and medium size software are usually divided into six stages that are interrelated with each other in a top-down approach named as waterfall. The input of one speciﬁc stage is the output of previous stage which initializes the next stage. At every step or stage of the model some additional information is also added up into the input of that stage and generating the results. But that additional information is restricted in scope and previous stages are directly traceable from that stage. The ﬁrst stage of the waterfall model is planning stage in which the most critical task is to clearly mention the high level requirements or goals of the software. It helps to make feasibility and risks associated with the project and also provides the basic project structure. The next stage is the requirement deﬁnition stage that takes goals as input from the planning stage and then these goals are deﬁned into a set of one or more requirements. Major functionalities, initial data entities and the operational data areas are deﬁned under this stage.

The input of design stage is the output of the requirement stage which is an approved requirements documentation. In this stage the design elements are deﬁned with the help of interviews, prototypes and workshops conducted. These design elements consist of functional hierarchy, business process diagrams, pseudo code and entity-relationship diagram. These design elements provide detail description about the software and each element is related to a speciﬁc requirement. The development stage is initialized by the previous design stage. At this stage the code for the design elements of software is written and it provides the functional software components. Integration and test stage is provided the output of the development stage. At this stage the diﬀerent software functioning components developed in the previous stage are integrated with each other to provide full ﬂedge software project which is providing all the high-level requirements. This software is also passed through diﬀerent test cases to check the validity, correctness, completeness and hence enables us to achieve our goals regarding our project. Finally, the installation and acceptance stage arrives. The software is loaded to the server at site of customers and tested with diﬀerent test cases once again to check the correct working. If all this shows adequate results satisfying the customer then the software is handed over to the customer formally.

**Reasoning to Choose Waterfall Model:-**

As in this model all the phases are in a sequence and are dependent with one another, therefore a phase cannot be started until the previous phase is completed and fully documented. This approach is most appropriate for our project, because all the requirements and goals of our project are very clear. Secondly, it is easy to do work in components and waterfall model is providing this approach. After the completion of all the phases individually, they are integrated together. This model is very economical and risk free due to its sequential approach.



## Fig. WaterFall SDLC Model Steps

## **Design Objectives**

The primary objective of the design of course, is to deliver the requirements as specified in the feasibility reports. In general the following design objectives should be kept in mind.

## **Practicality:-**

The system must be stable and can be operated by people with average.

## **Efficiency:-**

This involves accuracy, timeliness and comprehensiveness to the system output.

**Cost:-**

It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

**Flexibility:-**

The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems**.**

## **Security:-**

This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse.

System design involves first logical design and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, database and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer. When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system designing phase:

* 1. **Flowchart**

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it.

There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart area processing step, usually called activity, and denoted as a rectangular box a decision, usually denoted as a diamond. A flowchart is described as "cross-functional" when the page is divided into different swim lanes describing the control of different organizational units. A symbol appearing in a particular "lane" is within the Control of that organizational unit. This technique allows the author to locate the responsibility for performing an action or making a decision correctly, showing the responsibility of each organizational unit for different parts of a single process. Flowcharts depict certain aspects of processes and they are usually complemented by other types of diagram. For instance, Kaoru Ishikawa defined the flowchart as one of the seven basic tools of quality control, next to the histogram, Pareto chart, check sheet, control chart, cause-and-effect diagram, and the scatter diagram. Similarly, in UML, a standard concept-modeling notation used in software development, the activity diagram, which is a type of flowchart, is just one of many different diagram types.Nassi-Shneiderman diagrams and Dracon-charts are an alternative notation for process flow.

Common alternative names include: flow chart, process flowchart, functional flowchart, process map, process chart, functional process chart, business process model, process model, process flow diagram, work flow diagram, business flow diagram. The terms "flowchart" and "flow chart" are used interchangeably.

|  |  |  |
| --- | --- | --- |
| **Shape** | **Name** | **Description** |
|  | flow line | |  | | --- | | An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to. The line for the arrow can be solid or dashed. The meaning of the arrow with dashed line may differ from one flowchart to another and can be defined in the legend**.** | |  | |
|  | Terminals | Represented as circles, ovals, stadiums or rounded (fillet) rectangles. They usually contain the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product" |
|  | Process | Represented as rectangles. This shape is used to show that something is performed. Examples: "Add 1 to X", "replace identified part", "save changes", etc.... |
|  | Decision | Represented as a diamond (rhombus) showing where a decision is necessary, commonly a Yes/No question or True/False test. The conditional symbol is peculiar in that it has two arrows coming out of it, usually from the bottom point and right point, one corresponding to Yes or True, and one corresponding to No or False. | |
|  | Input/output | Represented as a parallelogram. Involves receiving data and displaying processed data. Can only move from input to output and not vice versa. Examples: Get X from the user; display X. | |
|  | Preparation | Represented as a hexagon. May also be called initialization. Shows operations which have no effect other than preparing a value for a subsequent conditional or decision step. Alternatively, this shape is used to replace the Decision Shape in the case of conditional looping. | |
|  | **Predefined Process** | Representedas rectangles with double-struck vertical edges; these are used to show complex processing steps which may be detailed in a separate flowchart. Example: process-files. One subroutine may have multiple distinct entry points or exit flows (see [coroutine](https://en.wikipedia.org/wiki/Coroutine)). If so, these are shown as labeled 'wells' in the rectangle, and control arrows connect to these 'wells'. | |

* 1. **Data Flow Diagram**

A Data Flow Diagram(DFD)is a diagram that describes the flow of data and the

Processes that change or transform data throughout a system. It’s a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information oriented and process oriented system flowcharts.

When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model. In the course of developing a set of leveled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model. Data flow diagrams can be used in both Analysis and Design phase of the SDLC.There are different notations to draw data flow diagrams (Yourdon & Coad and Gane &Sarsen, defining different visual representations for processes, data stores, data flow, and external entities

* + - **Zero Level DFD**

Book PackagesAdd Packages

View Bookings Log in

Assessor

Company HR

Log in Edit Packages

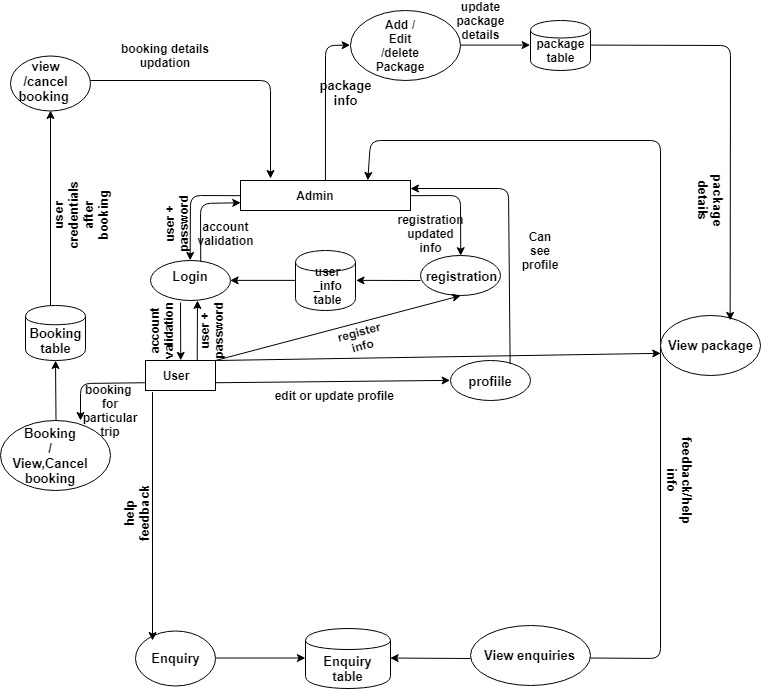
Update Profile Update Profile

RegistrationCancel Bookings

Enquire View Profile

View Tour detailsView Enquires

* + - **Level-1 DFD**



* 1. **Entity Relationship Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

## **Common Entity Relationship Diagram Symbols:**

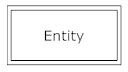
An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

* **Entities**, which are represented by rectangles. An entity is an object or concept about

Which you want to store information.

.

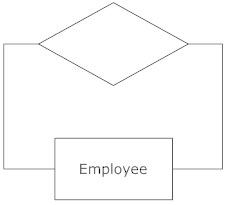
* A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



* **Actions**, which are represented by diamond shapes, show how two entities share information in the database



* .In some cases, entities can be self-linked. For example, employees can supervise other employees.



* **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



* A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values

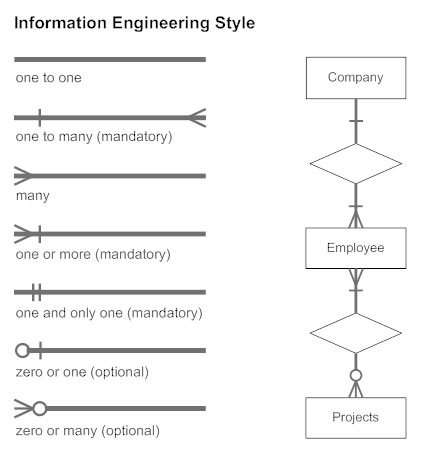
.

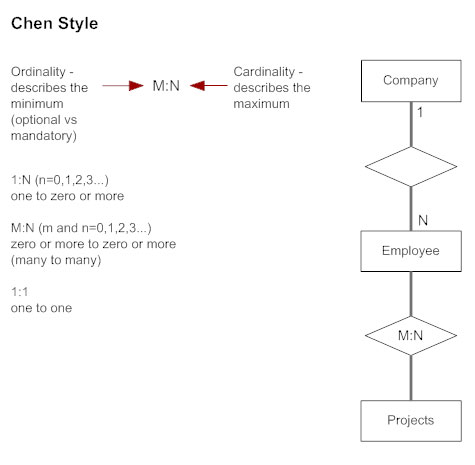
* A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary

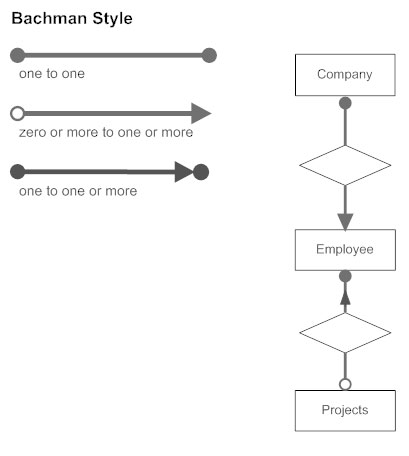
.

* **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
* **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

There are many notation styles that express cardinality.  
**Information Engineering Style**

****

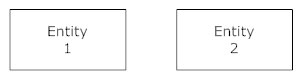
**ChenStyle  
**

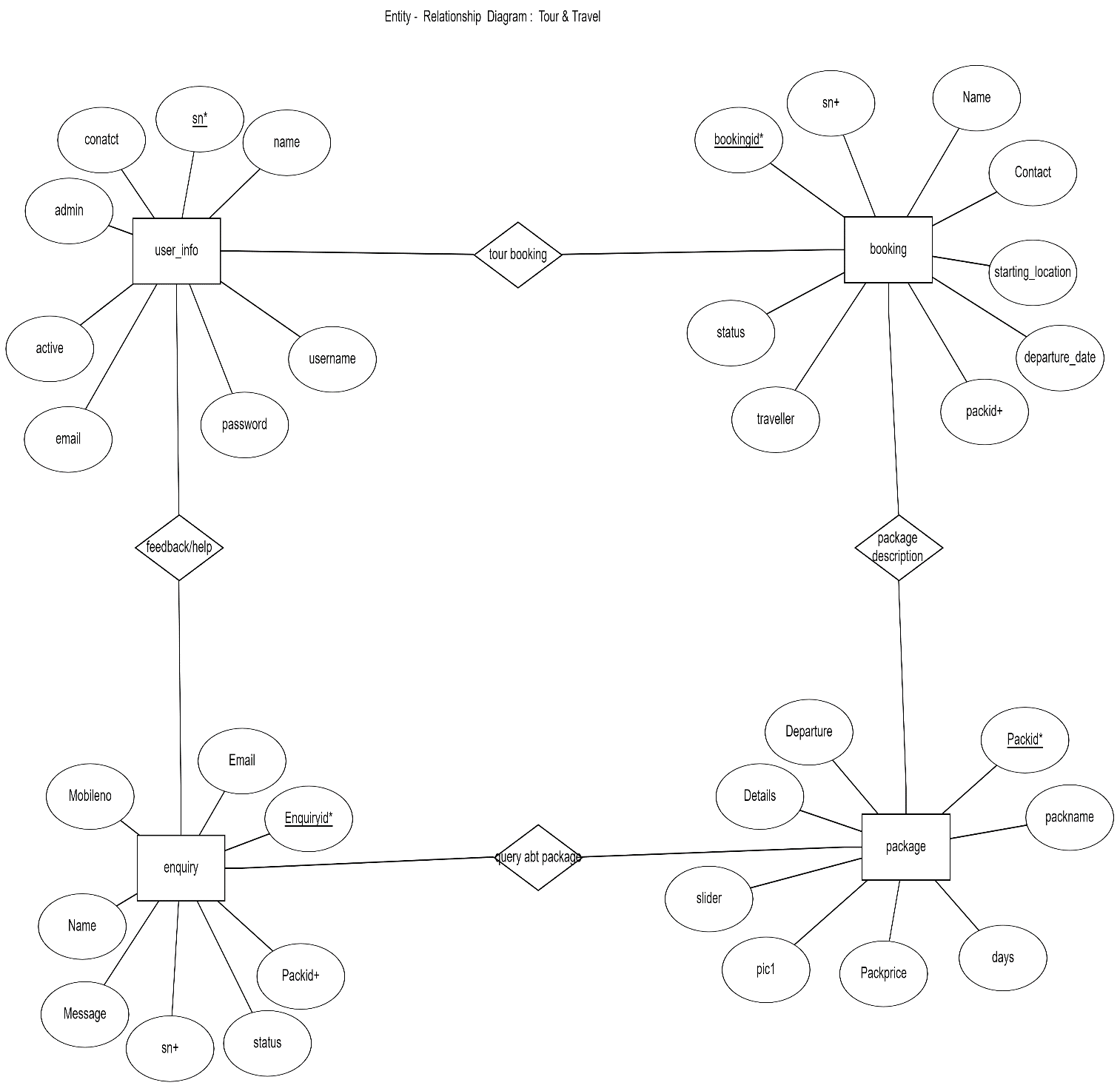
****

## **ER Diagram Uses:-**

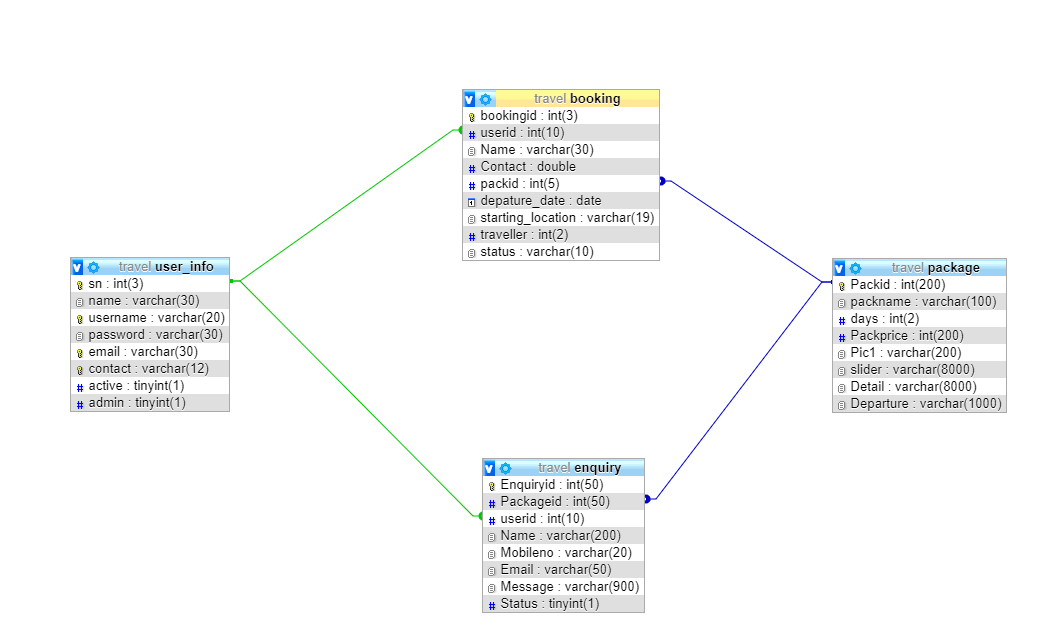
When documenting a system or process, looking at the system in multiple ways increases the understanding of that system. ERD diagrams are commonly used in conjunction with a data flow diagram to display the contents of a data store. They help us to visualize how data is connected in a general way, and are particularly useful for constructing a relational database.

**Here are some best practice tips for constructing an ERD:**

* Identify the entities. The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.  
  
* Identify relationships. Look at two entities, are they related? If so draw a solid line connecting the two entities.
* Describe the relationship. How are the entities related? Draw an action diamond between the two entities on the line you just added. In the diamond write a brief description of how they are related.
* Add attributes. Any key attributes of entities should be added using oval-shaped symbols.
* Complete the diagram. Continue to connect the entities with lines, and adding diamonds to describe each relationship until all relationships have been described. Each of your entities may not have any relationships, some may have multiple relationships. That is okay.
  + 1. **Logical Entity Relationship Diagram**



* + 1. **Physical Entity Relationship Diagram**

****

**Use Case Diagram**

User

Admin

1. INPUT/OUTPUT FORM DESIGN
   1. **Input Design (Introduction):-**

Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the outputs. Input is one of the most expensive phases of the operation of a computerized system and creates sometimes a major problem. Different type of problem with a system can usually be traced back to faulty input design method needless to say, therefore, that the input data are the lifeblood of a system and have to be analyzed and designed with utmost care and consideration. Input design features can ensure the reliability of the system and generate correct reports form the accurate data. The input design also determines whether the user can interact efficiently with the system.

* 1. **Elements of Input Data:-**

In accurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing.

* 1. **Input Data:-**

The goal of designing input data is to make data entry as easy, logical and error free from errors as possible. In entering data, operators need to know the following:

* The allocated space for each field.
* Field sequence, which much match that in the source document.
* The format in which data fields are entered for example, filling out the date field is required through the edited format mm/dd/yy.

When we approach input data design, we design the source document. Let us elaborate on each step.

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%2220%22%20y%3D%2220%22%20width%3D%22120%22%20height%3D%2260%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3E

* 1. **Source Documents**

Source data are captured initially on original paper or a source document. For example, a cheque written against an account is a source document. When it reaches the bank, it is encoded with special magnetic ink character recognition so that a reader that is part of the information system of the bank can process it. Therefore, source documents initiate a processing cycle as soon as they are entered into the system. Source documents may be entered into the system from punch cards, from diskettes, or even directly through the keyboard.

A source document should be logical and easy to understand. Each area in the form should be clearly identified and should specify for the user what to write and where to write it. A source document may or may not be retained in the proposed system. Thus, each source document may be evaluated in terms of. It’s continued use in the proposed system.The extent of modification for the proposed system & Replacement by an alternative source document.

* 1. **Input Design Guidelines**

The design of input play very significant role in getting the correct output. It covers al phases of input from creation of initial data (original recording) to actual entering the data to the system for processing. The input design is the link that ties the information system into the world of its users. Some features of design may vary depending on whether the system is batch-oriented or on-line. Here, we will discuss the various objectives of input design. They focus on:

* Controlling amount of input
* Avoiding delay
* Avoiding errors in data
* Avoiding extra steps
* Keeping the process simple
  1. **Each of the five objectives of input design is briefly discussed below:**

**Controlling Amount of Data:-**

An effective design controls the quantity of data for input for the following reasons:  
Firstly, data preparation and data entry operations depend on people. Since labor costs are high, the cost or preparing and entering data is also high. It is quite evident, then that reducing data requirements mean lowering costs through reduced labor expense.  
Secondly, the input phase of computing can be slow process and take many times longer than that needed by computers to carry out their tasks. In fact, the computer itself may sit idle until data is prepared and input for processing. By reducing input requirements, the analyst will speed the entire process from data capture to processing to provide result to users.

**Avoiding Delay:-**

When processing is delayed owing to data preparation or data entry, the cause is called a bottleneck. Avoid bottlenecks when designing input should always be one of the objectives of the analyst.

**Avoiding Errors in Data:-**

The third objective deals with errors. In one sense, the rate at which errors occur is dependent on the quantity of data. Since the lower the amount of data is inputted, there are fewer opportunities for the error to occur.

Firstly, the analyst can reduce this number by reducing the volume of data dust must be entered for each transaction. Secondly, the analyst can also affect error rates of an operation through design. The manner in which data must be entered can reduce the chance of errors.

Still, a third aspect of error control is the need to detect errors when they do occur. Checks and balances in the data entry programs, called input validation techniques, also detect errors input.

1. **Database Design**
2. **User Details Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| u\_id | Int (10) |
| Name | Varchar (30) |
| Username | Varchar (30) |
| Password | Varchar (30) |
| Email | Varchar (30) |
| Contact | Varchar (30) |
| Active | Boolean |
| admin | Boolean |

1. **Booking Details Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| booking\_id | Int (5) |
| Userid | Int (10) |
| Name | Varchar (30) |
| Contact | Double |
| Packageid | Int (5) |
| Departure\_date | Date |
| Starting\_location | Varchar (30) |
| No\_travelers | Int (2) |
| Status | Boolean |

1. **Package Details Table**

|  |  |
| --- | --- |
| **Field Name** | **Data type** |
| Package\_id | Int (5) |
| Package\_Name | Varchar (30) |
| Duration | Int (2) |
| Package\_price | Int (200) |
| Image | Varchar (200) |
| Slider\_image | Varchar (2000) |
| Details | Varchar (8000) |
| Departure | Varchar (200) |

1. **Enquiry Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Enquiry Id | Int (10) |
| Package Id | Int (10) |
| User id | Int (4) |
| Name | Varchar (20) |
| Mobile No | Varchar (30) |
| Email | Varchar (30) |
| Message | Varchar (8000) |
| Status | Boolean |

1. Snapshots of Product

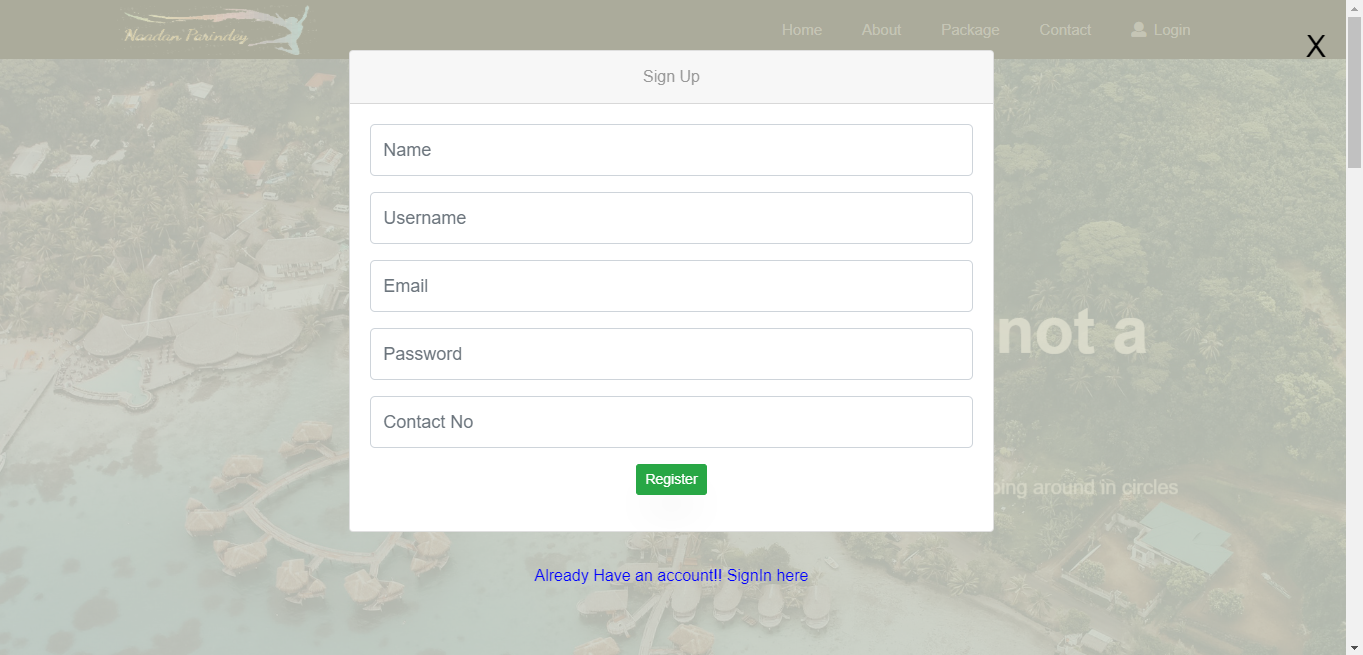


Fig 1: User Registration

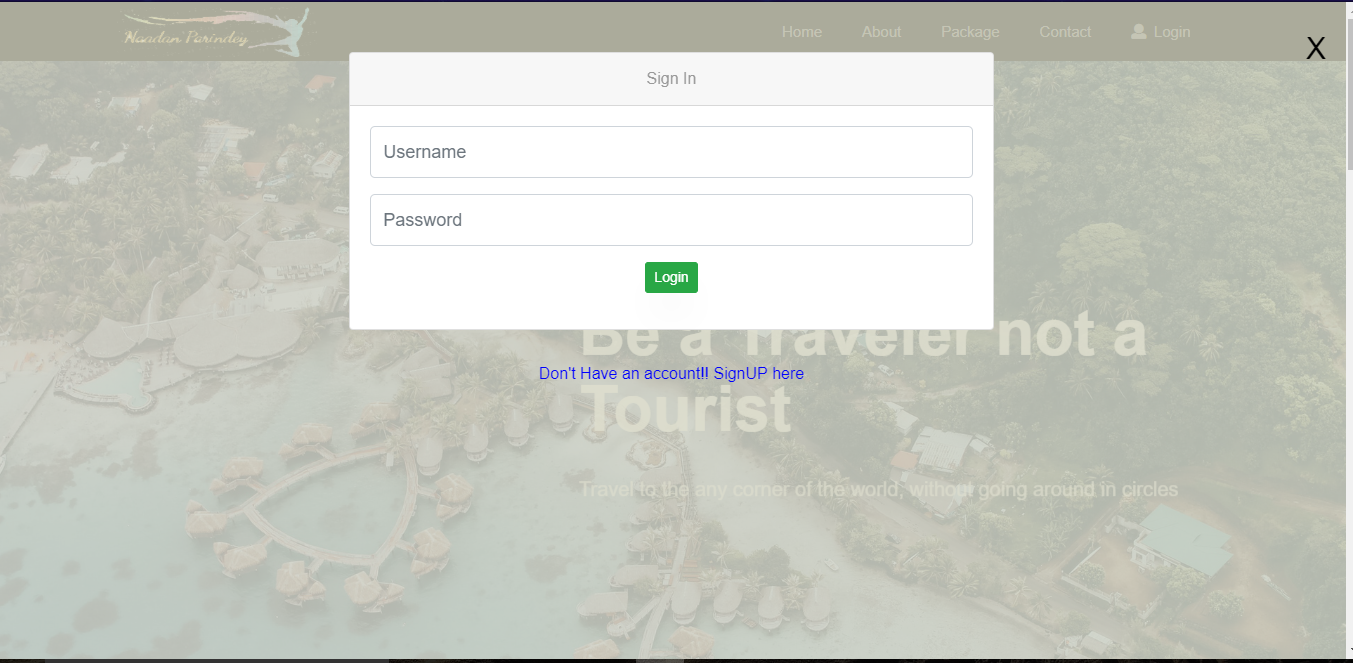


Fig 2: User Login

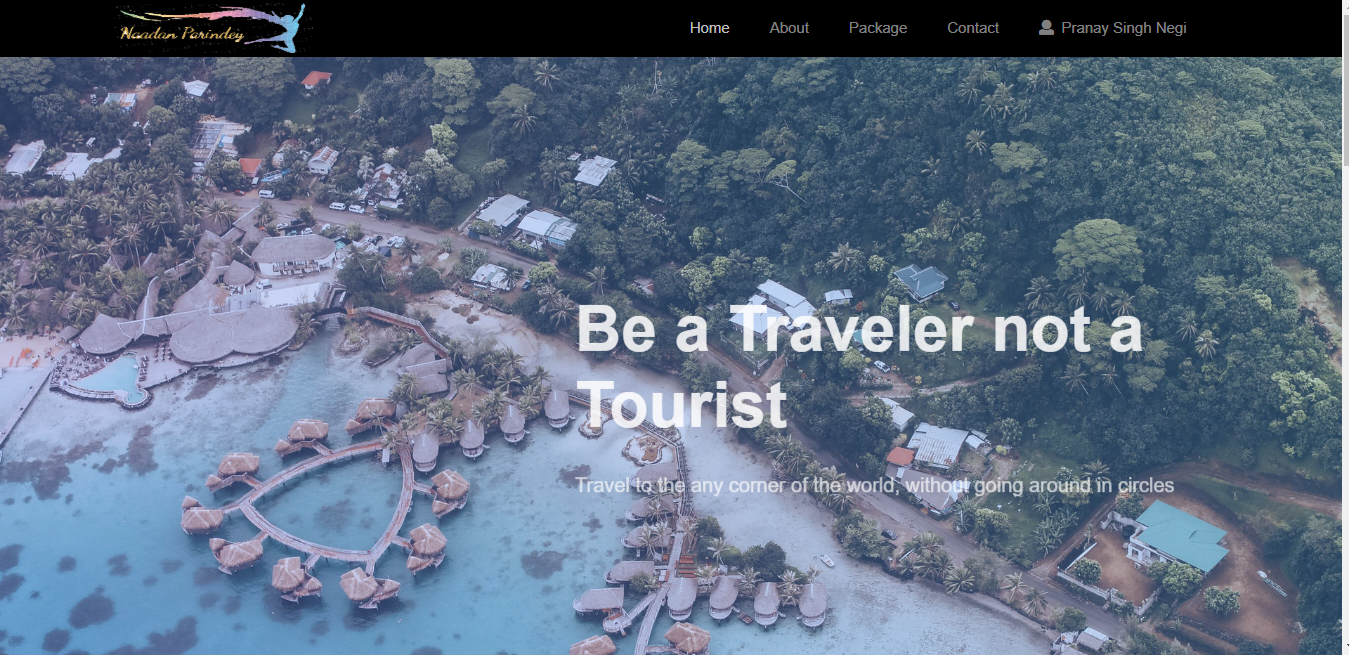


Fig 3: Home Page

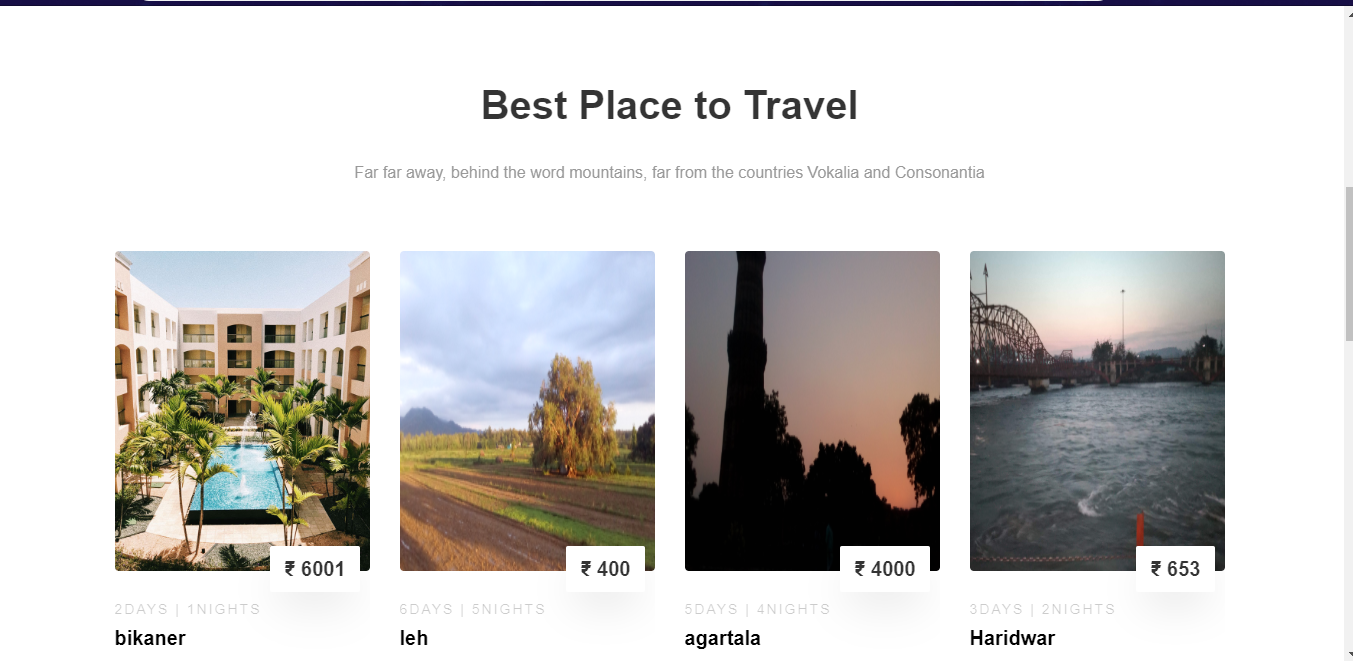


Fig 4: Tour Packages

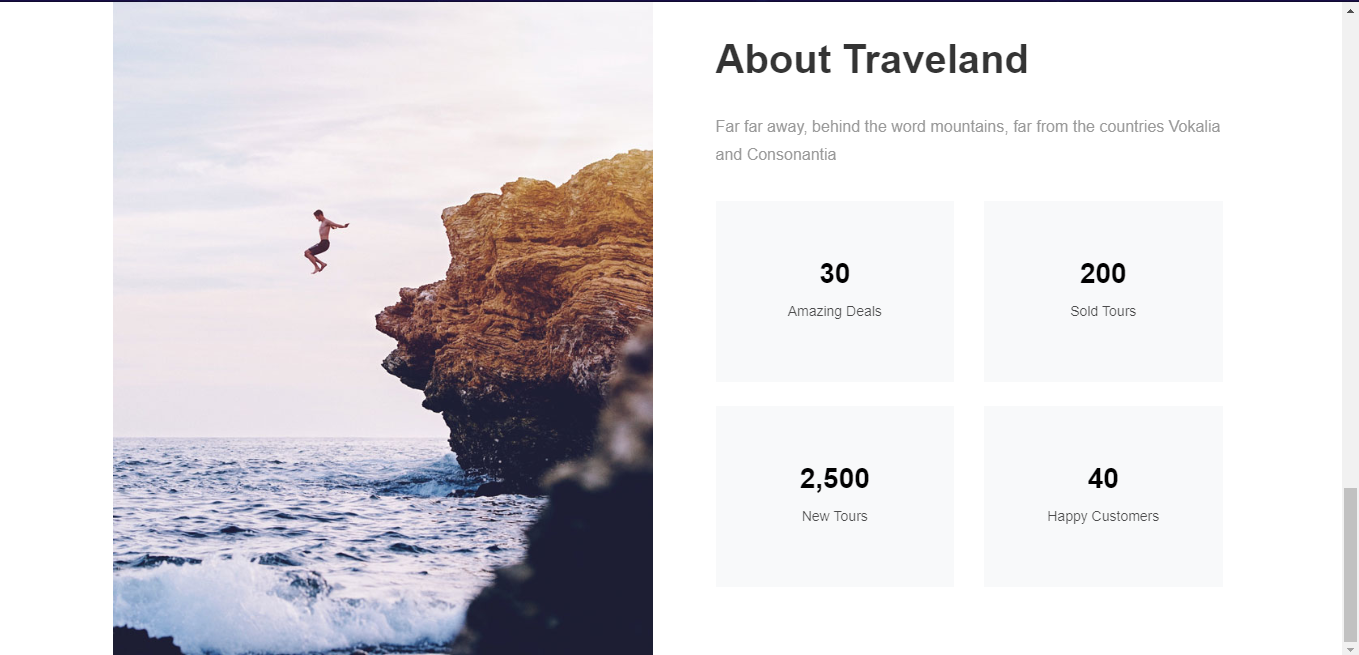


Fig 5: About Page

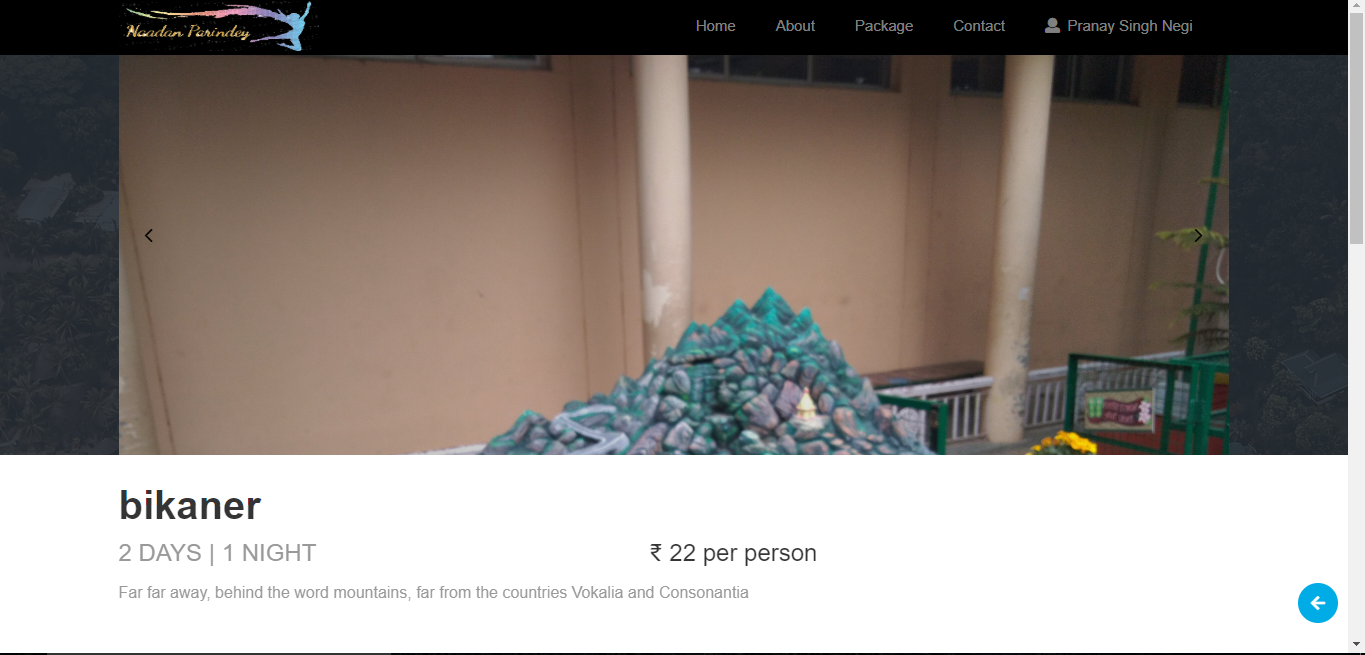


Fig 6: Package bookings (1)

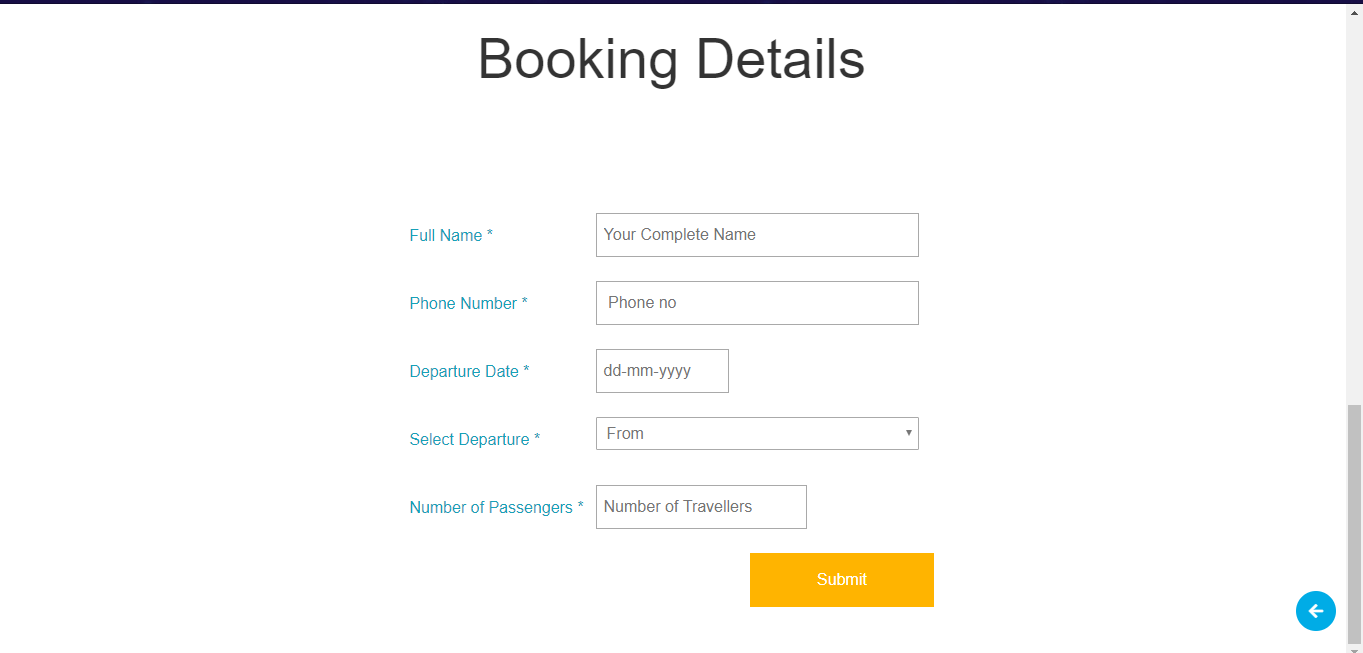


Fig. 7: Package booking (2)

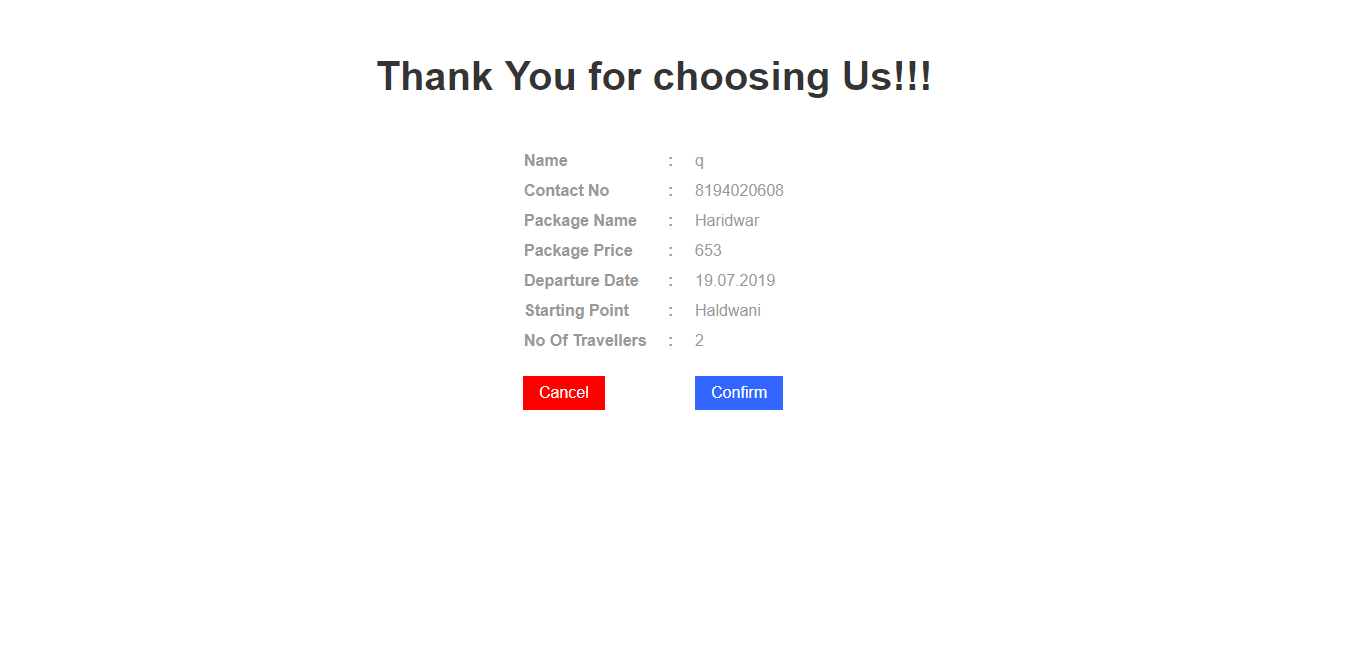


Fig 8: Booking Confirmation

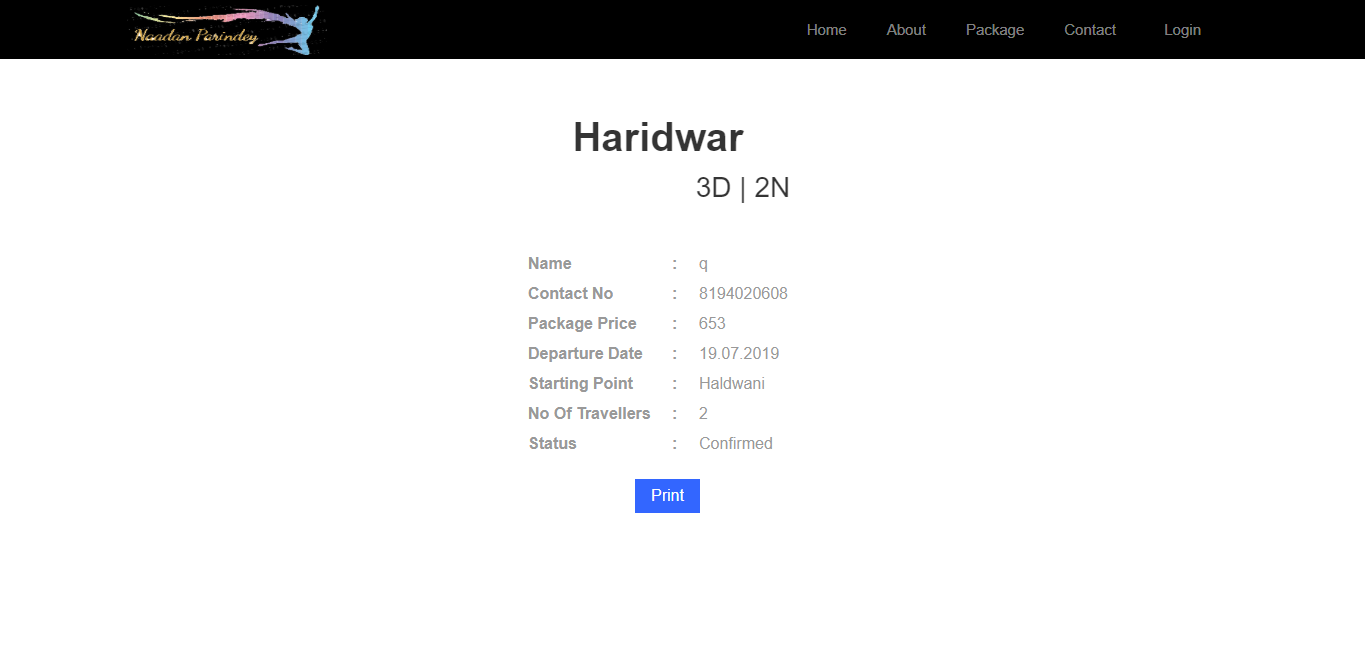


Fig 9: Print Ticket

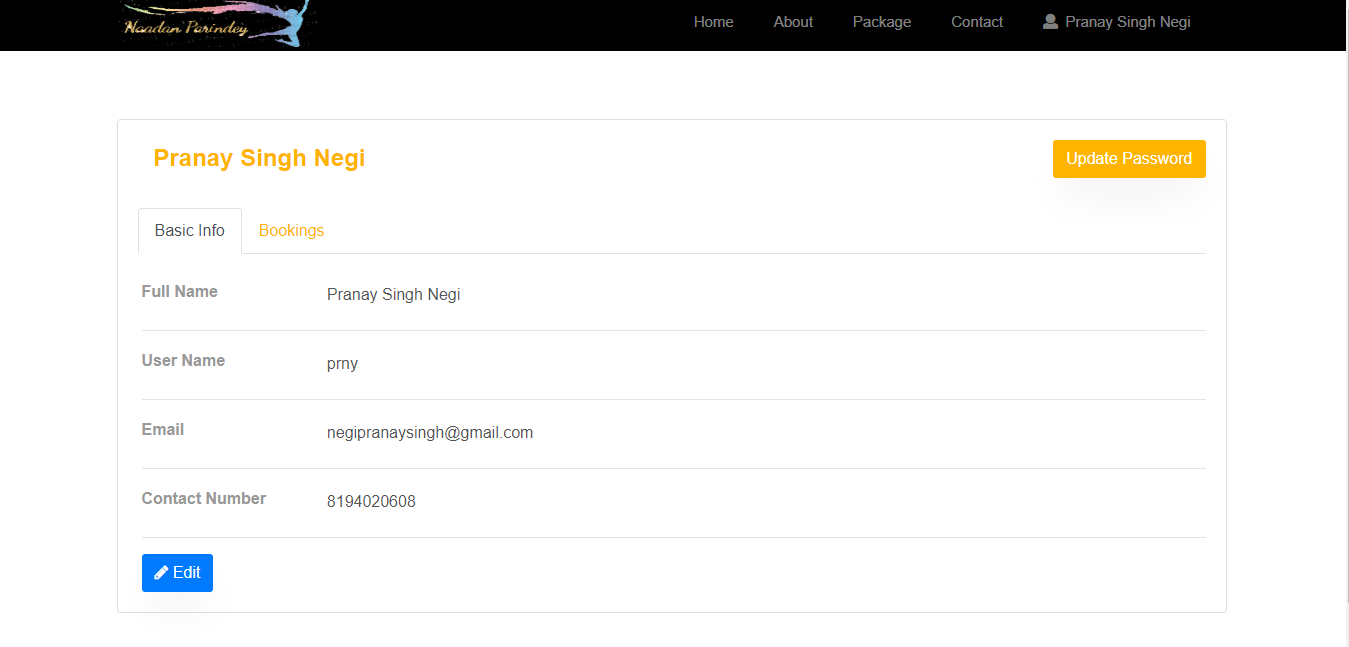


Fig 10: Profile

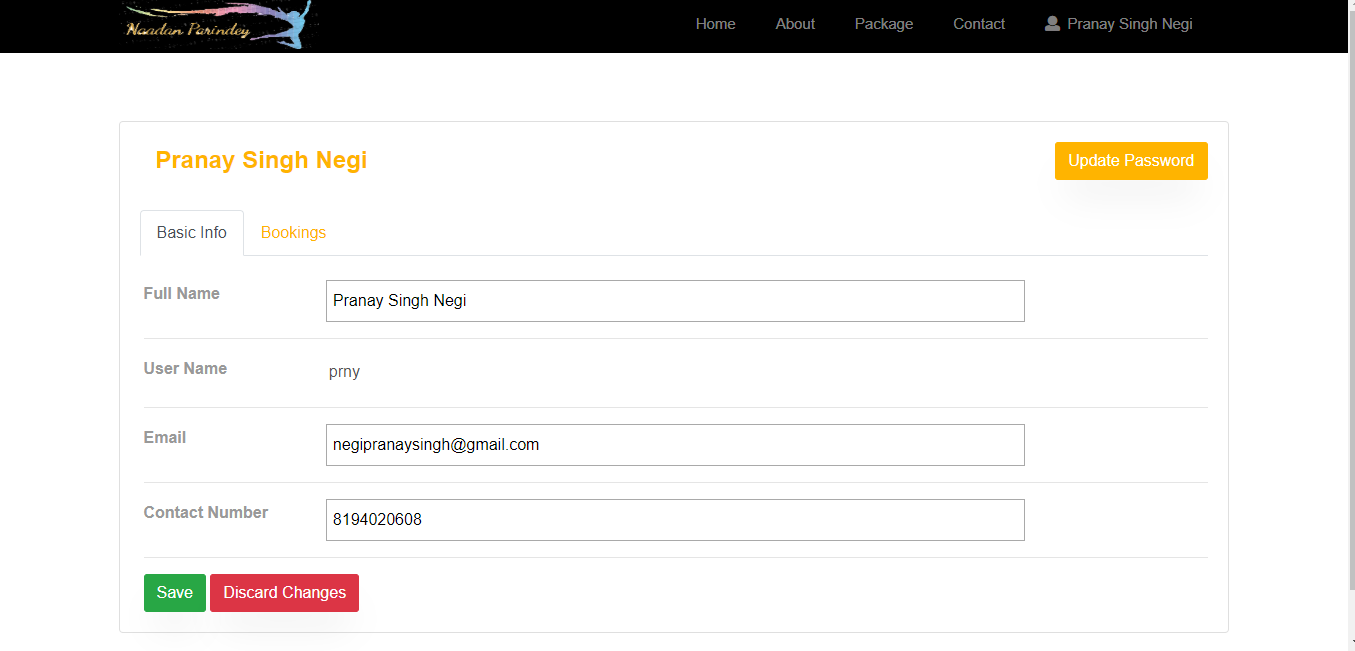


Fig 11: Edit Profile

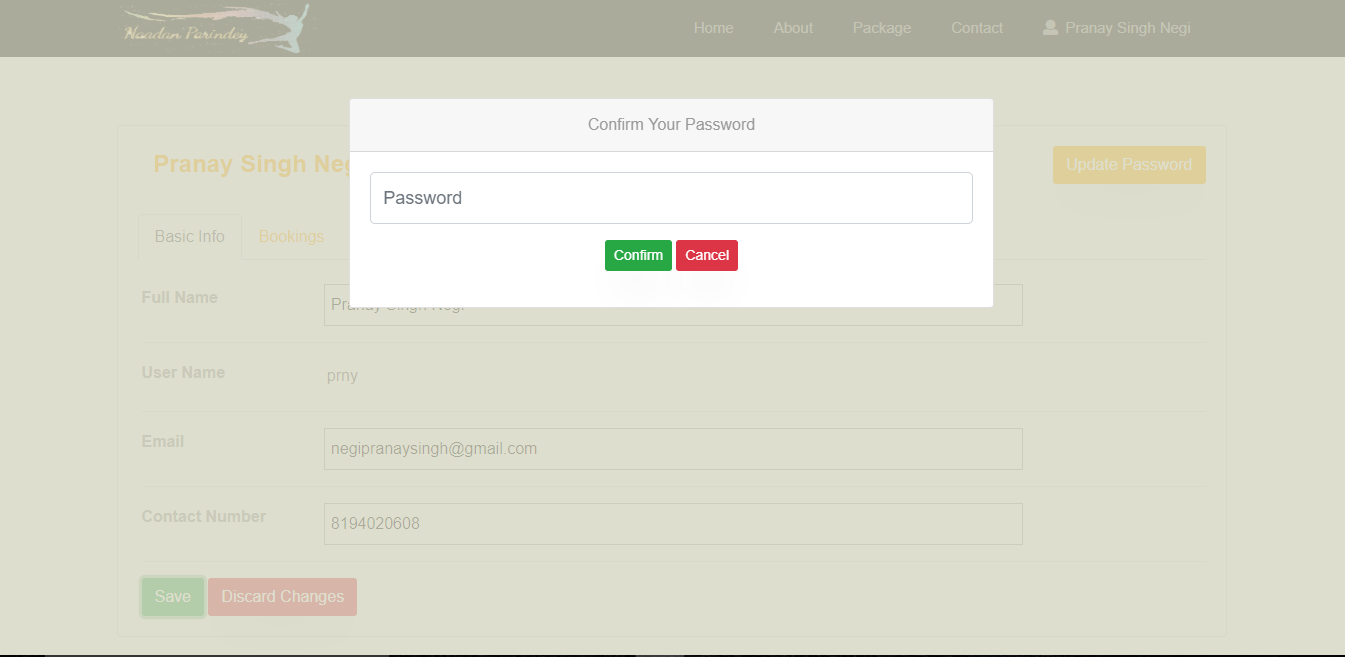


Fig 12: Account Verification for Updating Profile

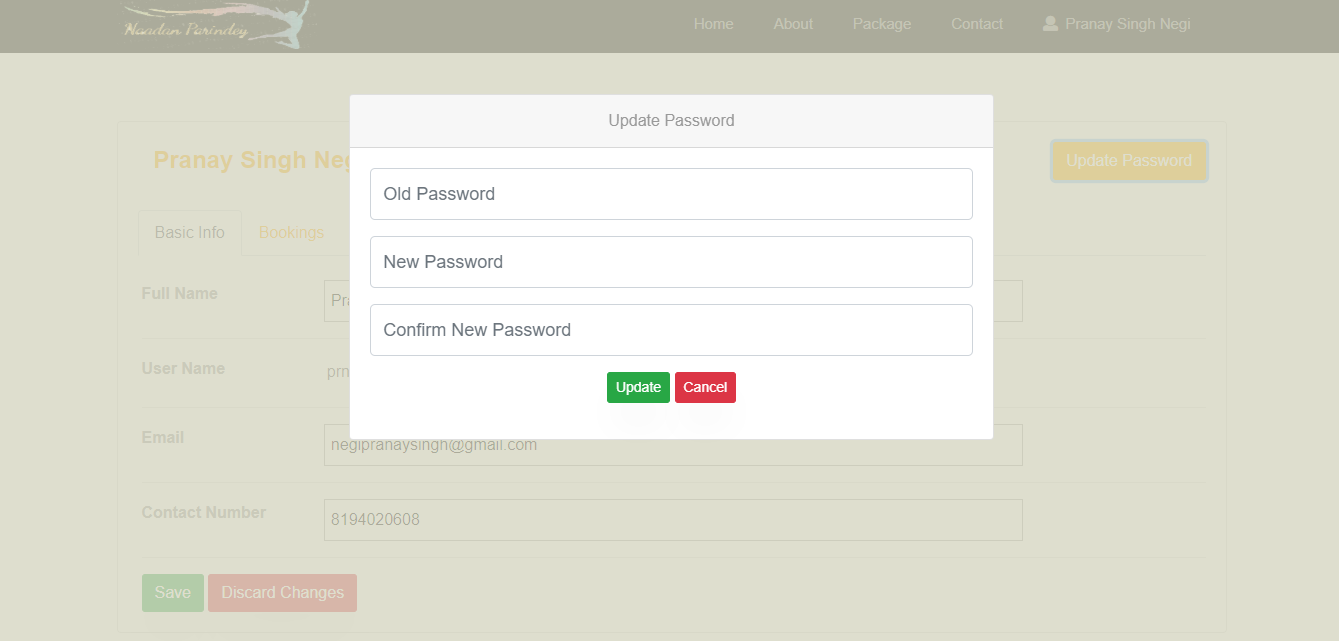


Fig 13: Update Password

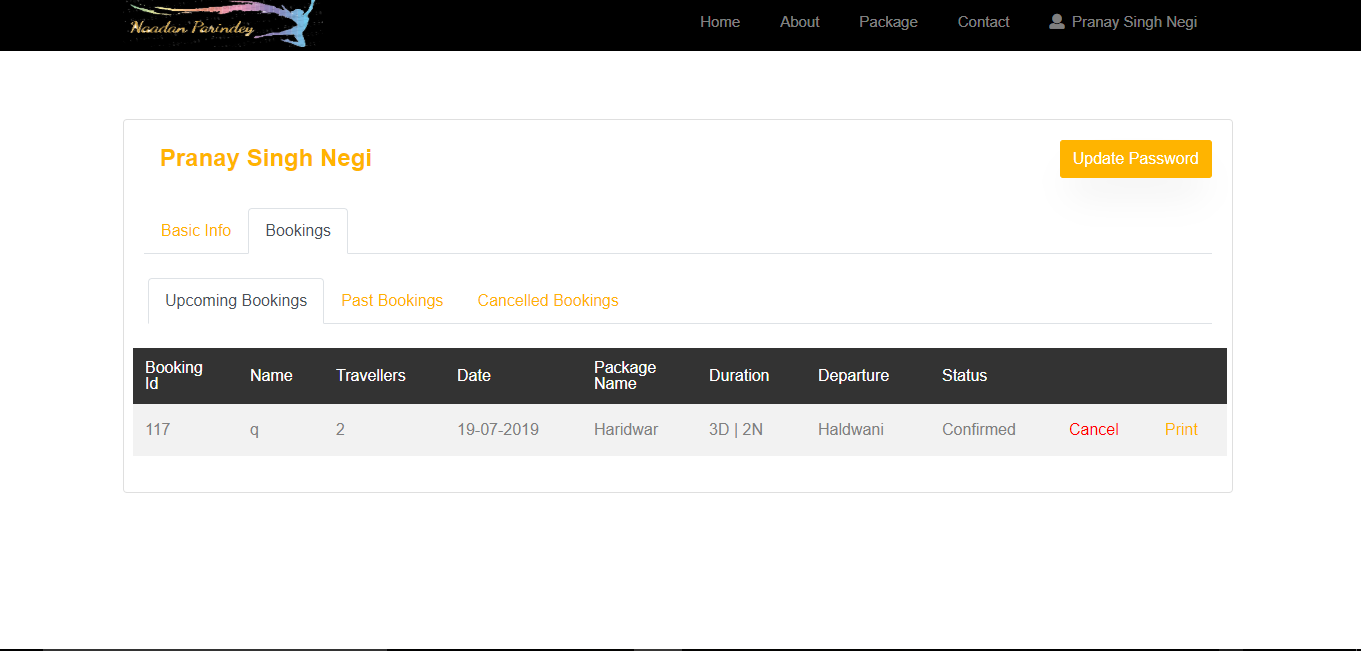


Fig 14: User’s Upcoming Bookings

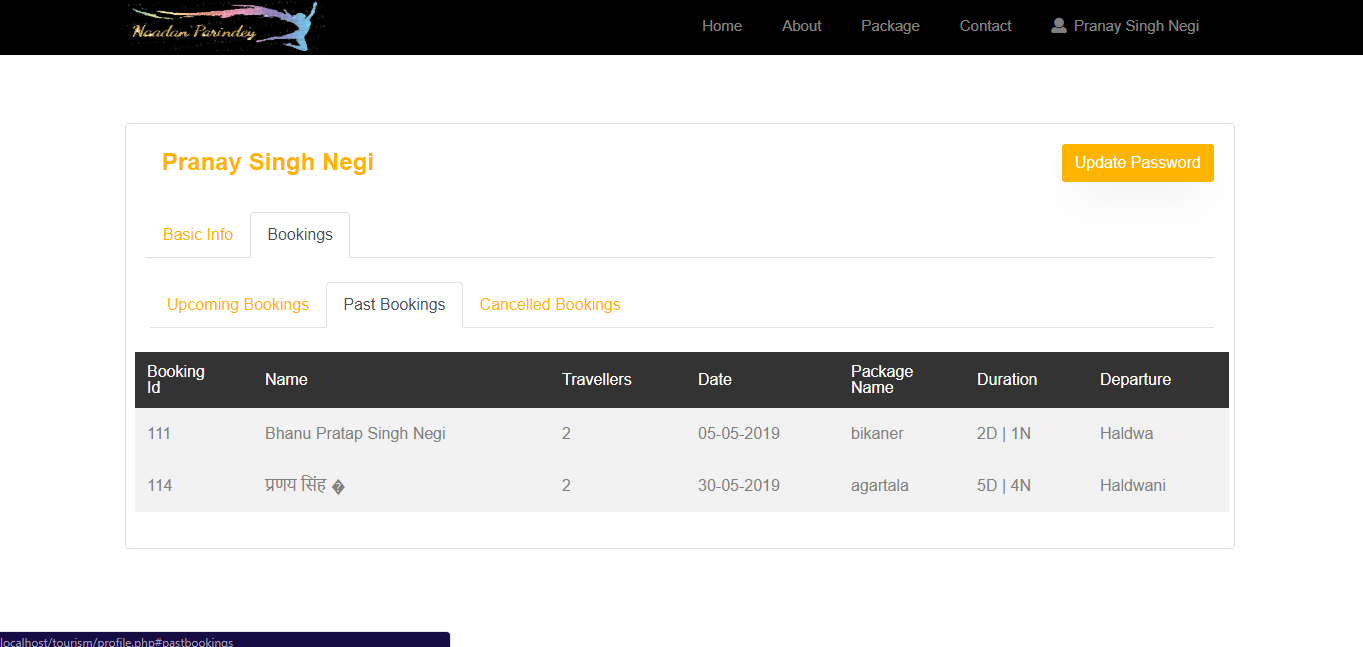


Fig 14: User’s Past Bookings

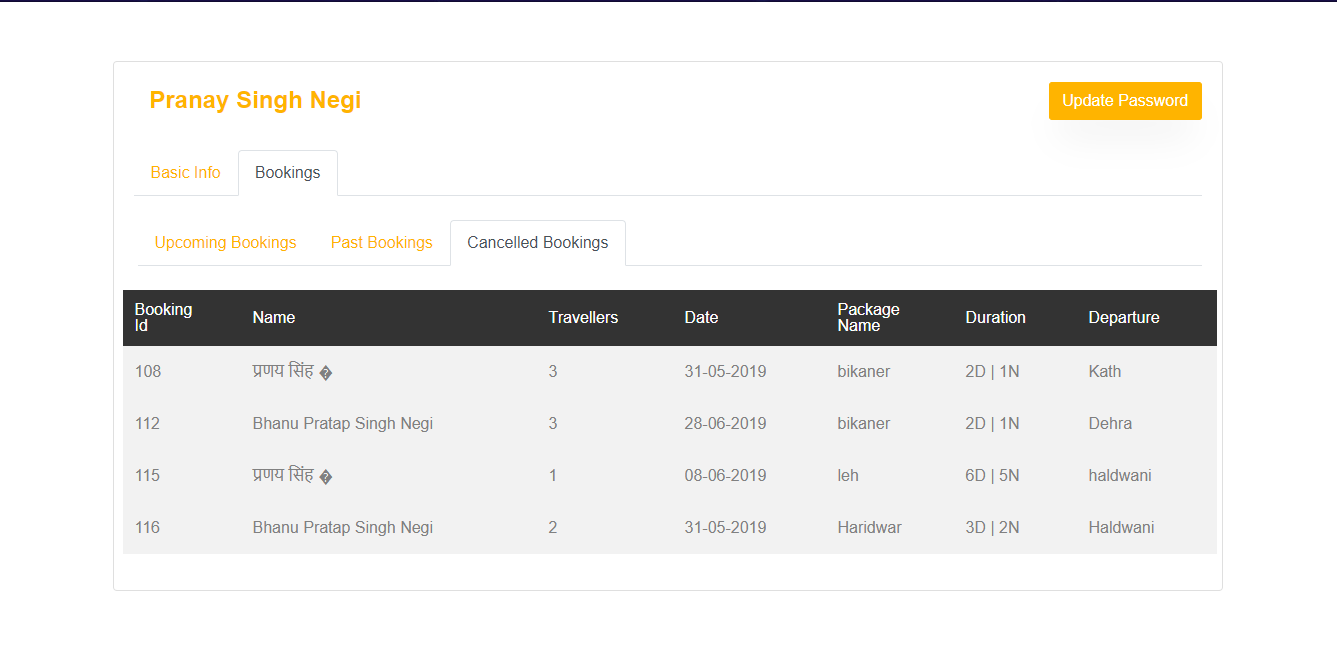


Fig 15: Cancelled Bookings

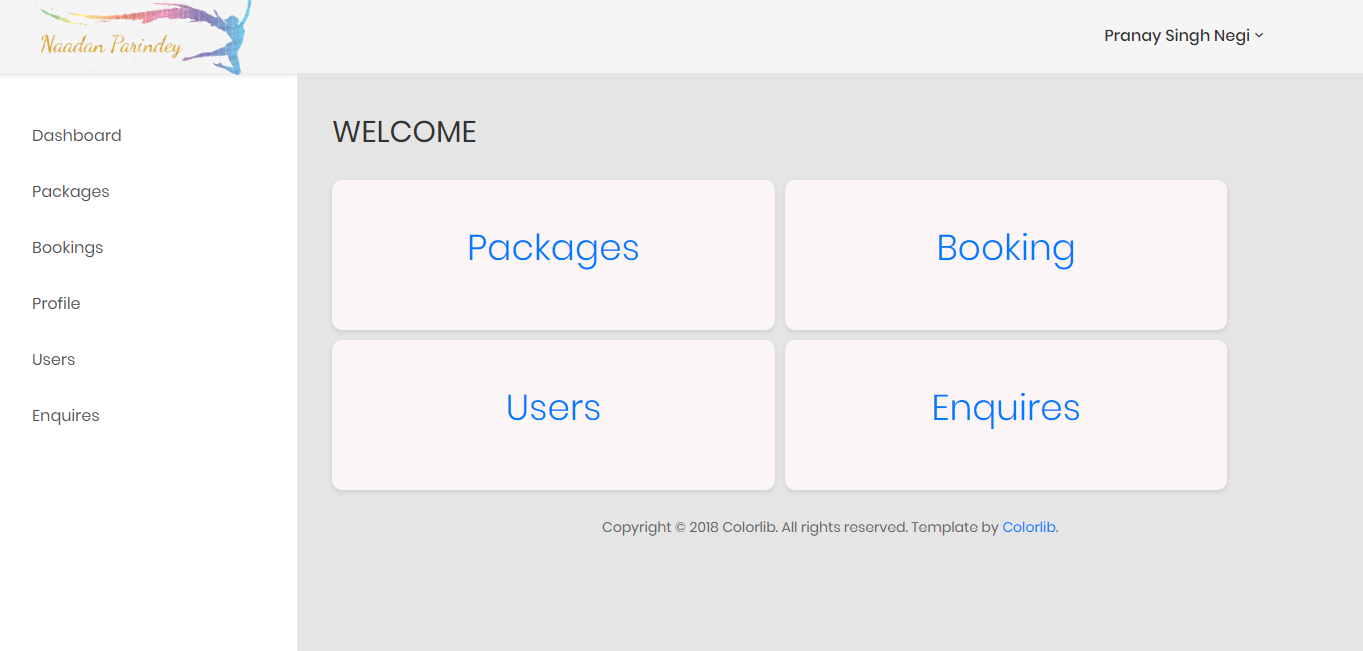


Fig 16: Admin Dashboard

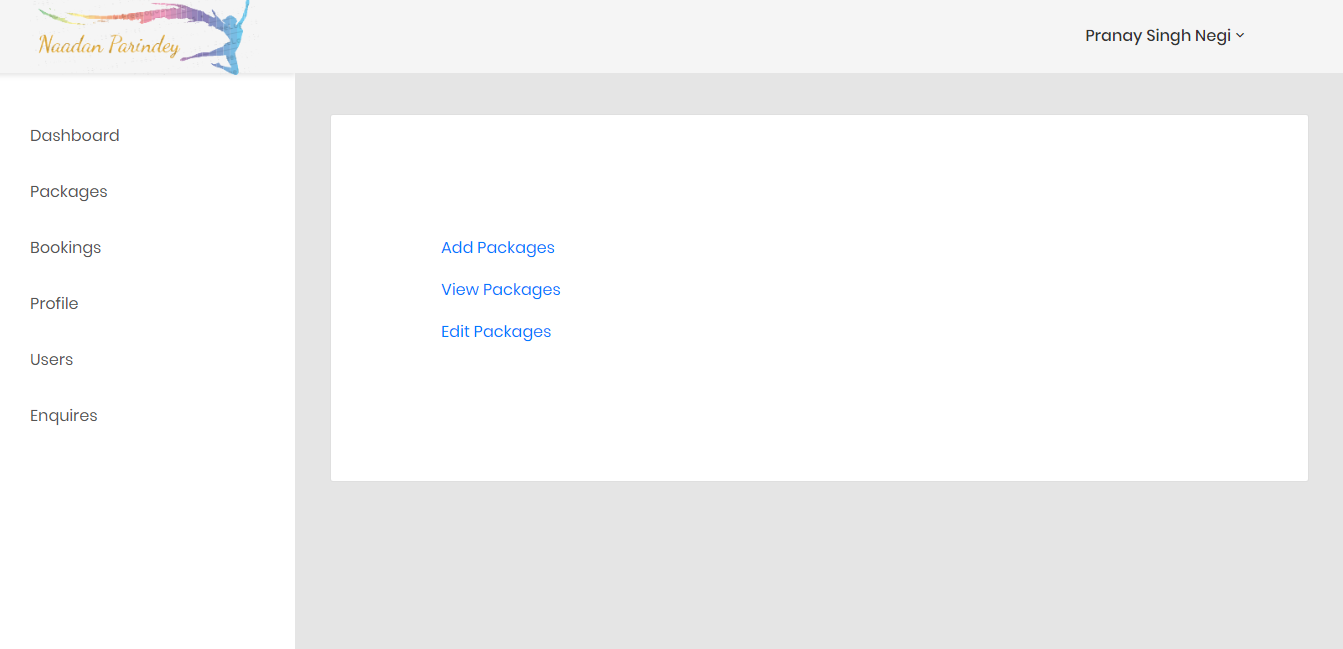


Fig 17: Package Page

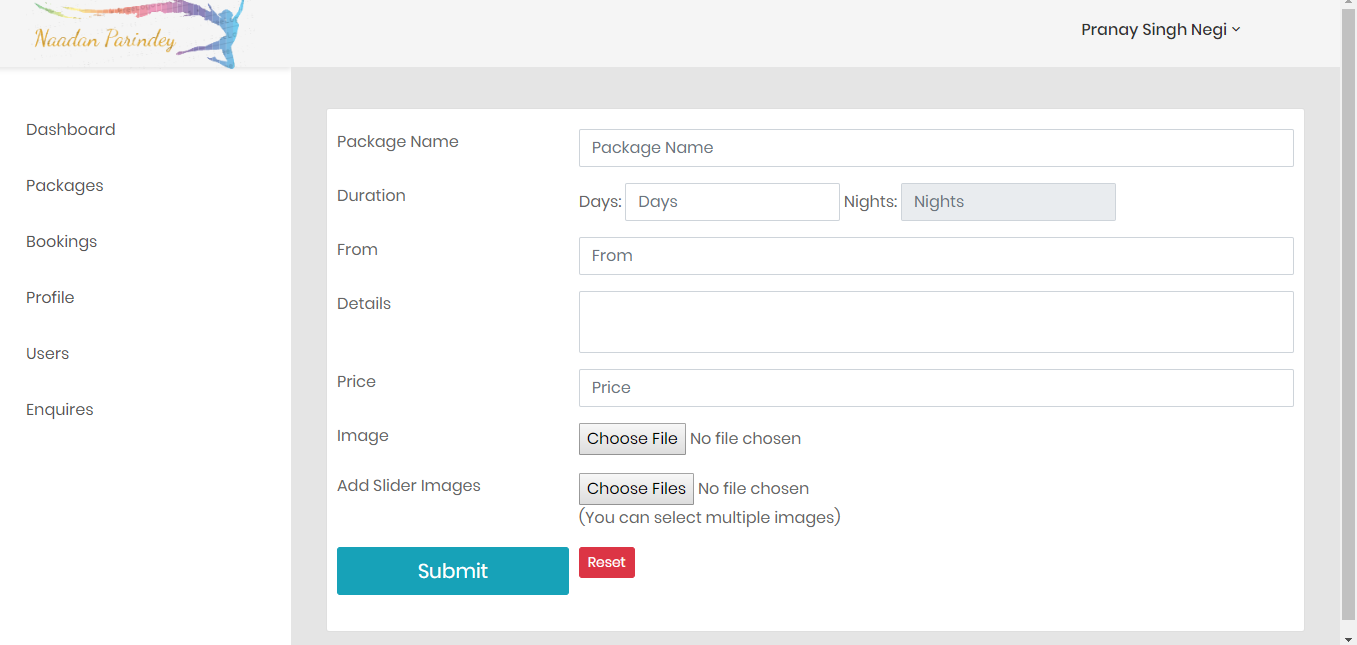


Fig 18: Add Package Form

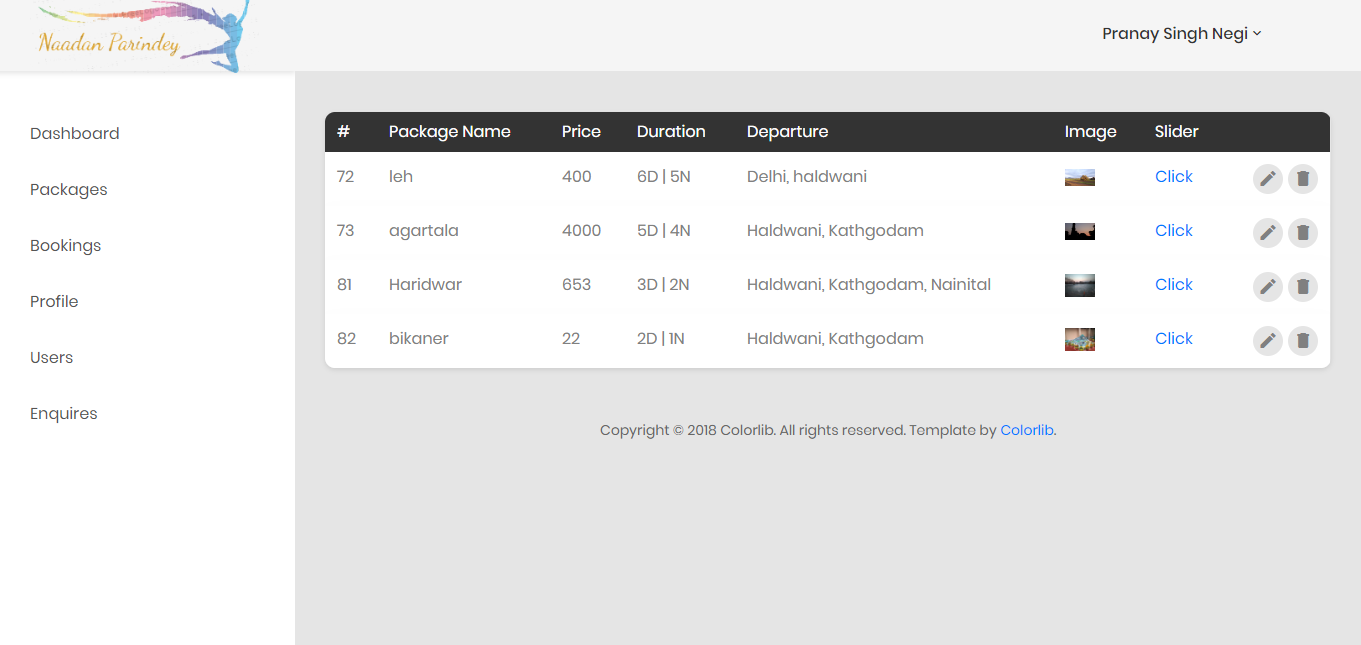


Fig 19: View Package

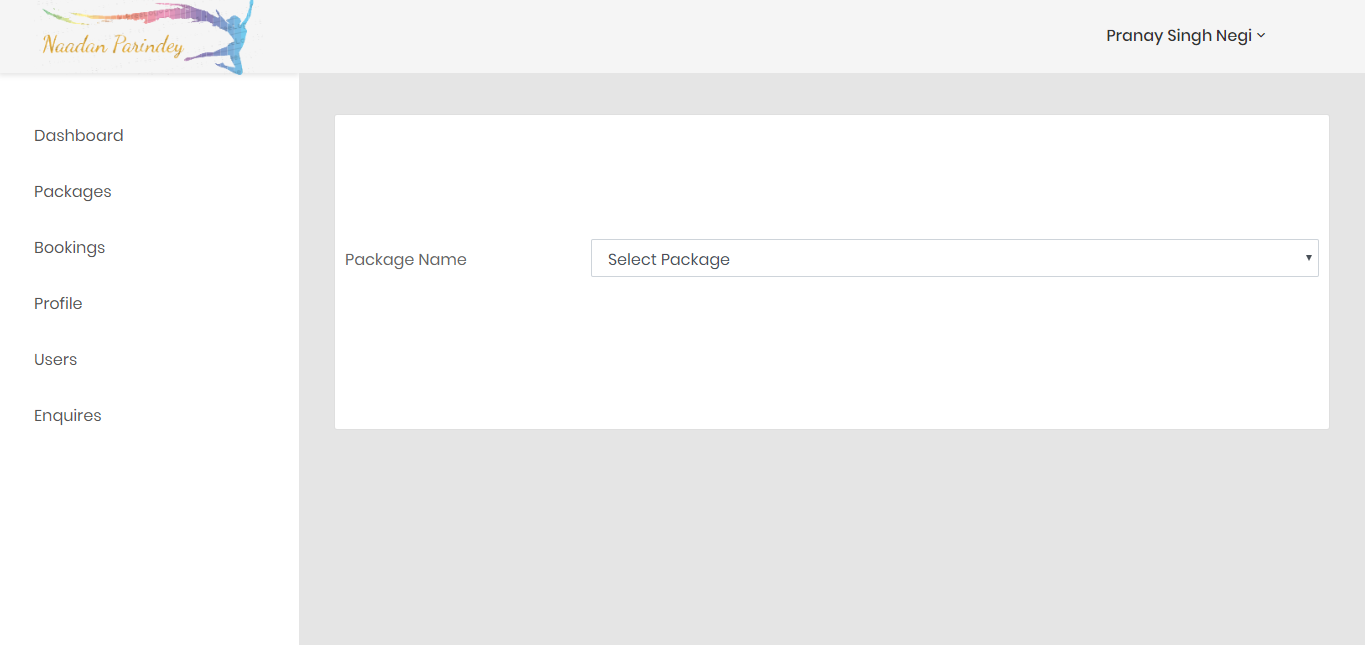


Fig 20: Select Package for Updation

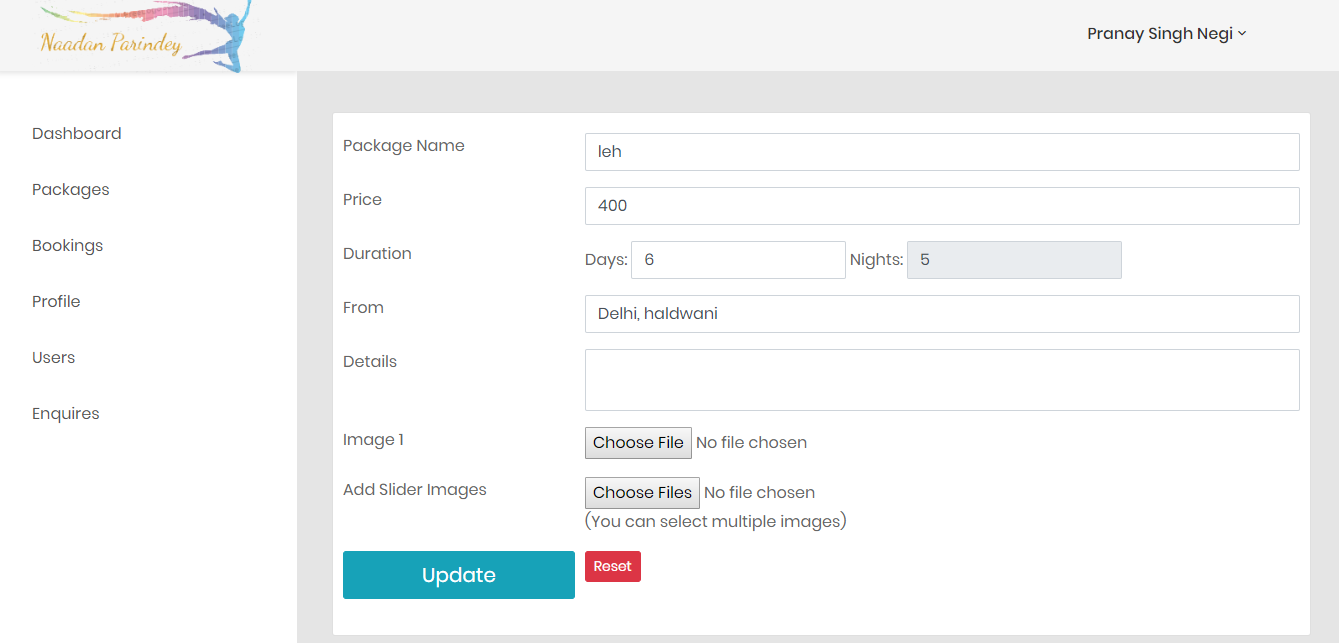


Fig 21: Edit Package Form

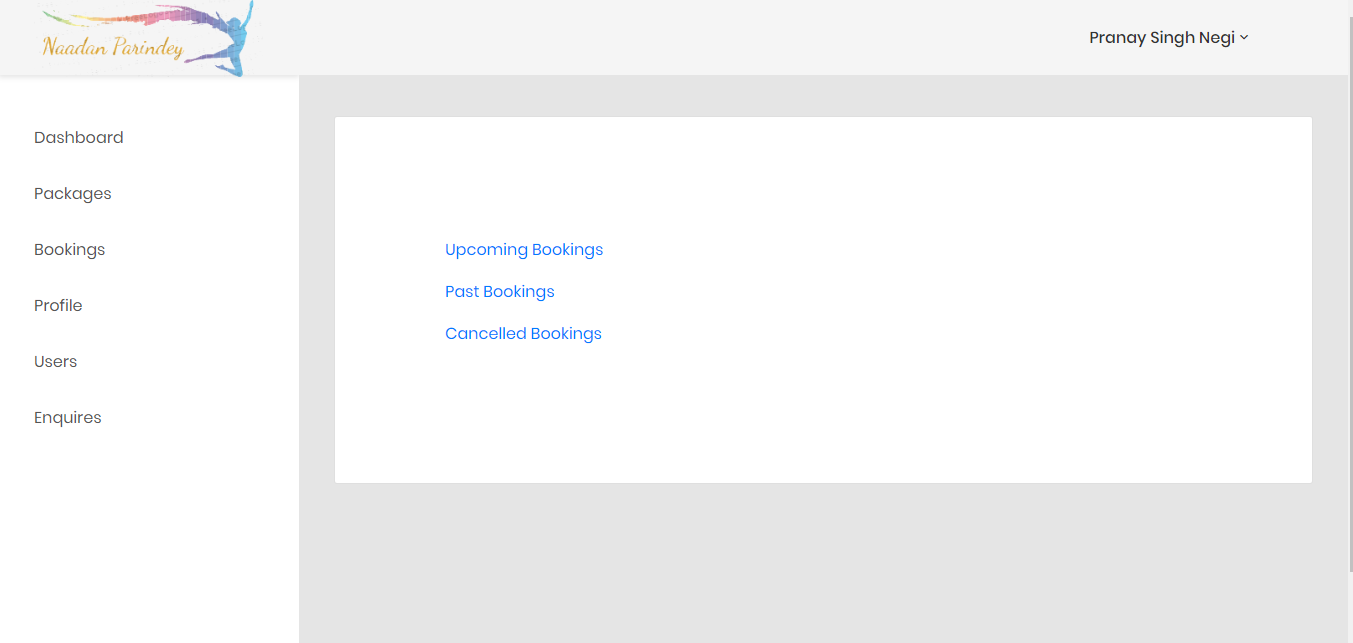


Fig 22: Bookings Page

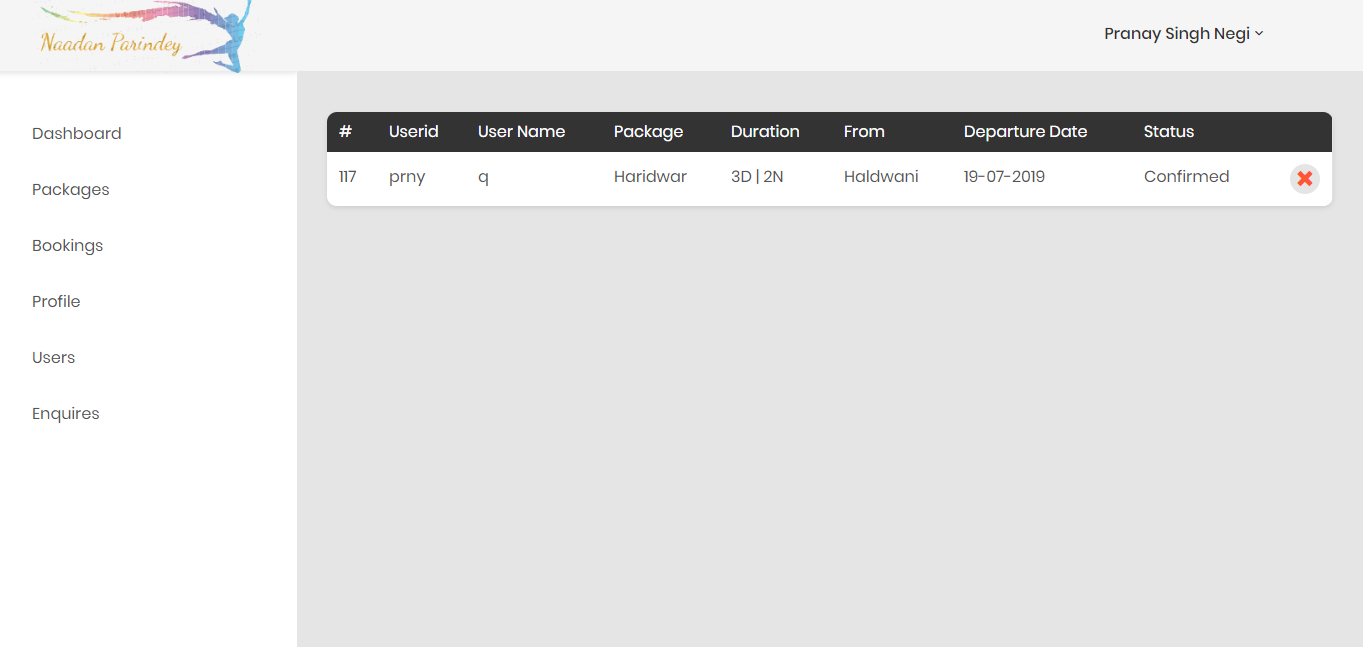


Fig 23: Upcoming Bookings

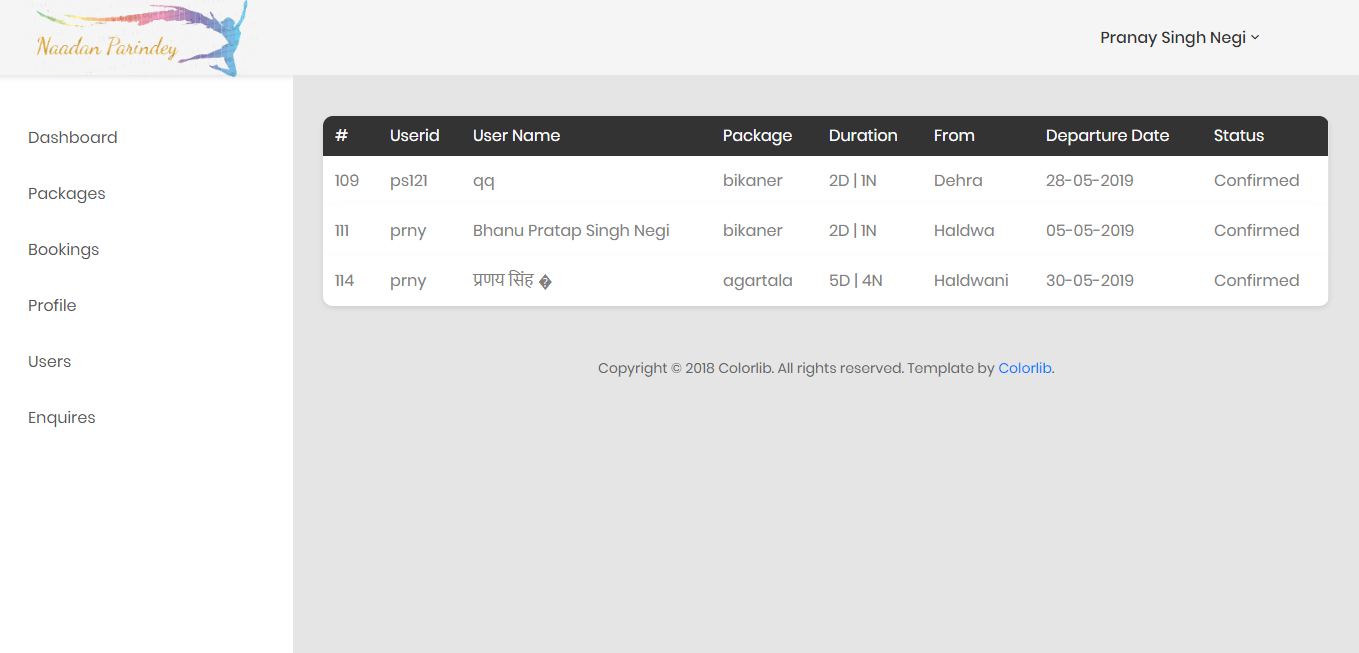


Fig 24: Past Bookings

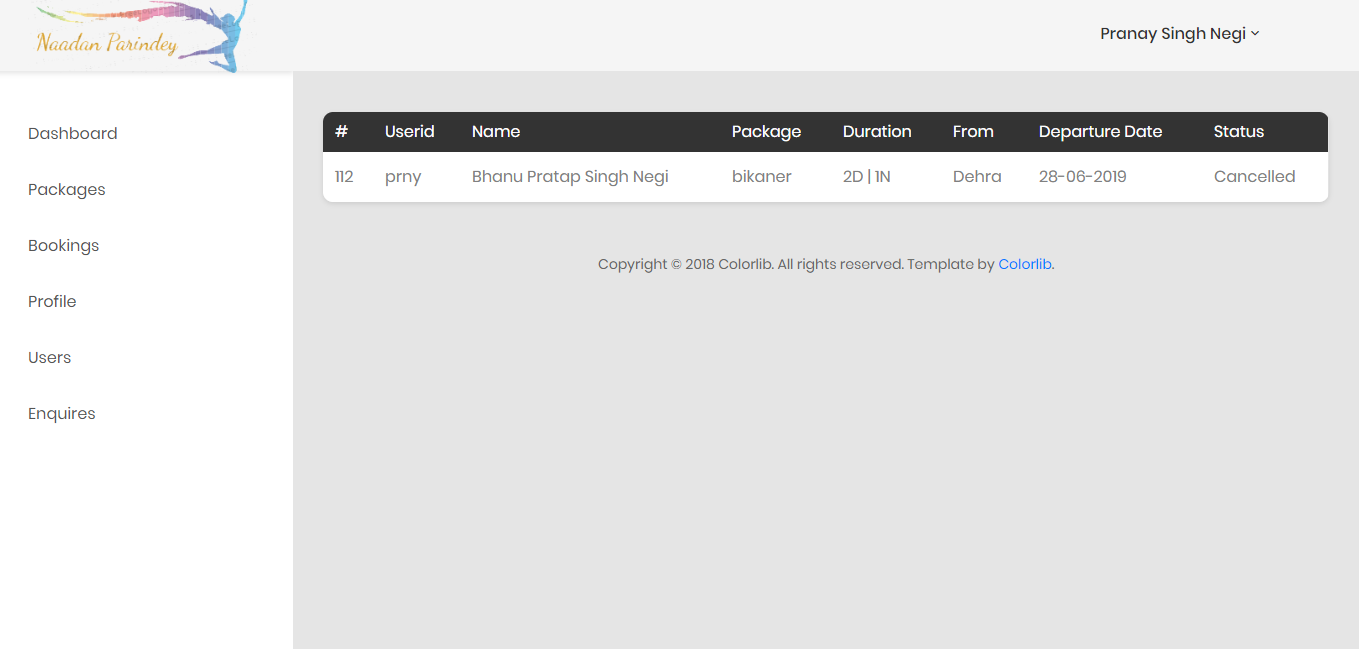
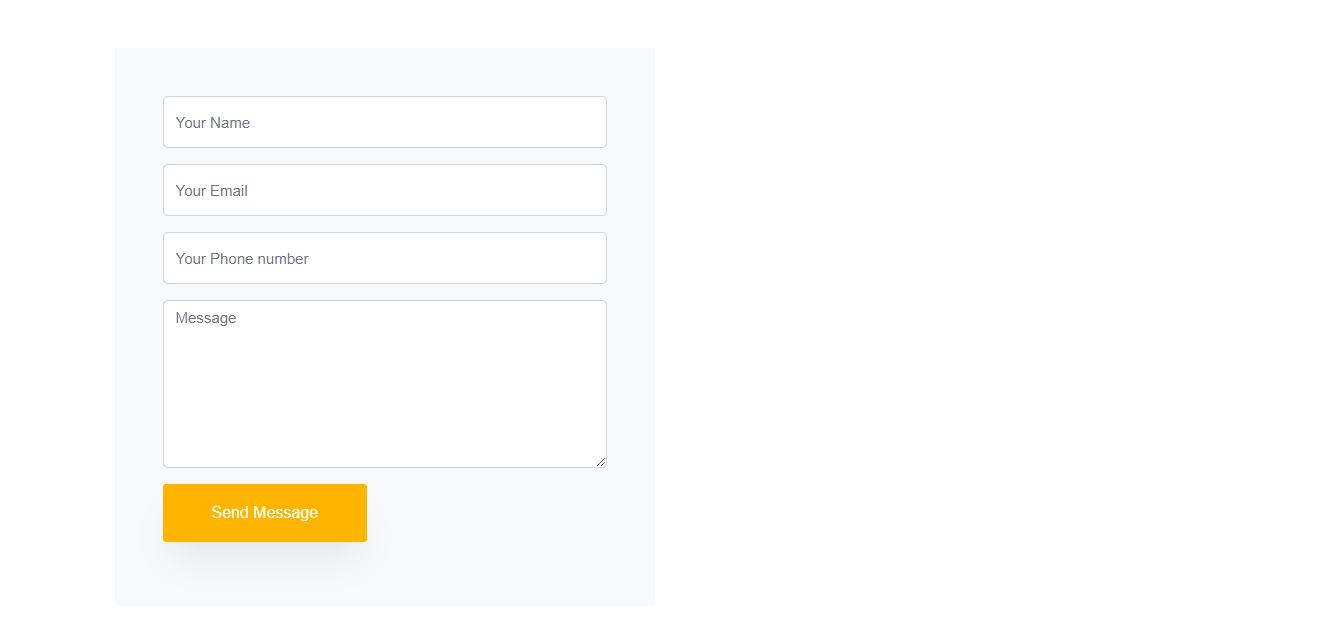


Fig 25: Cancelled Bookings

Fig 26: Enquiry Form (Users)

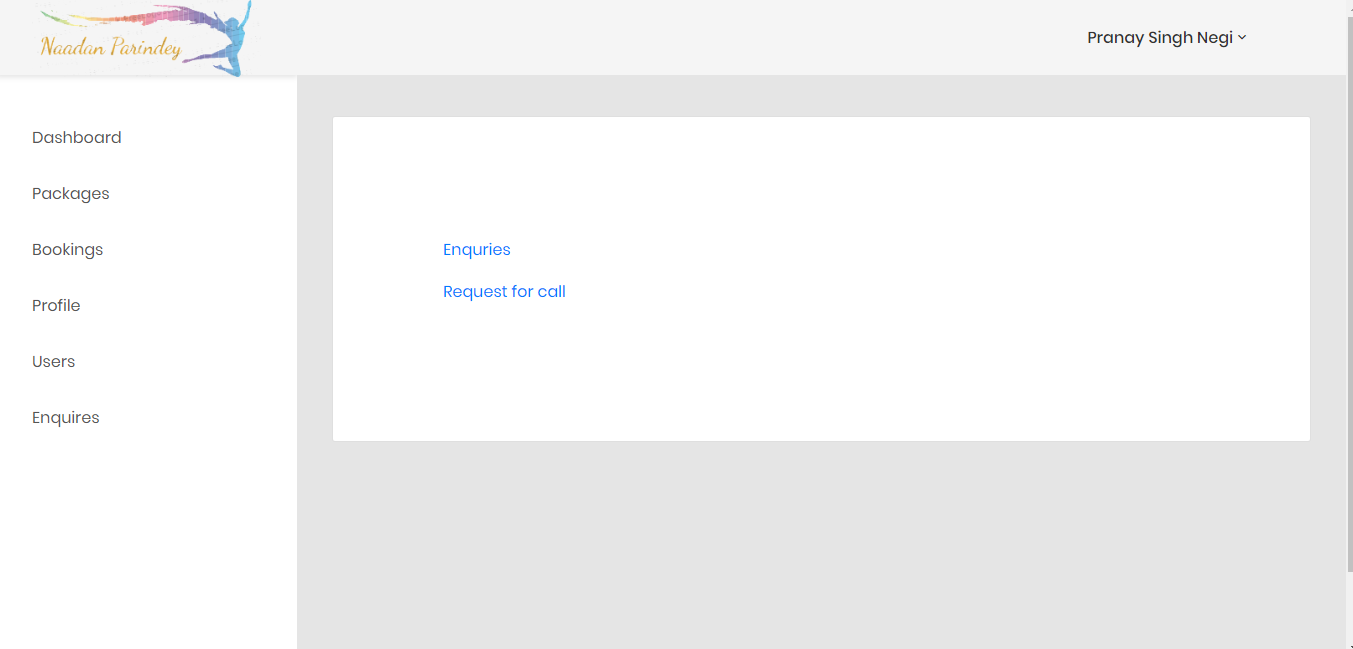


Fig 27: Enquiry View (Admin)

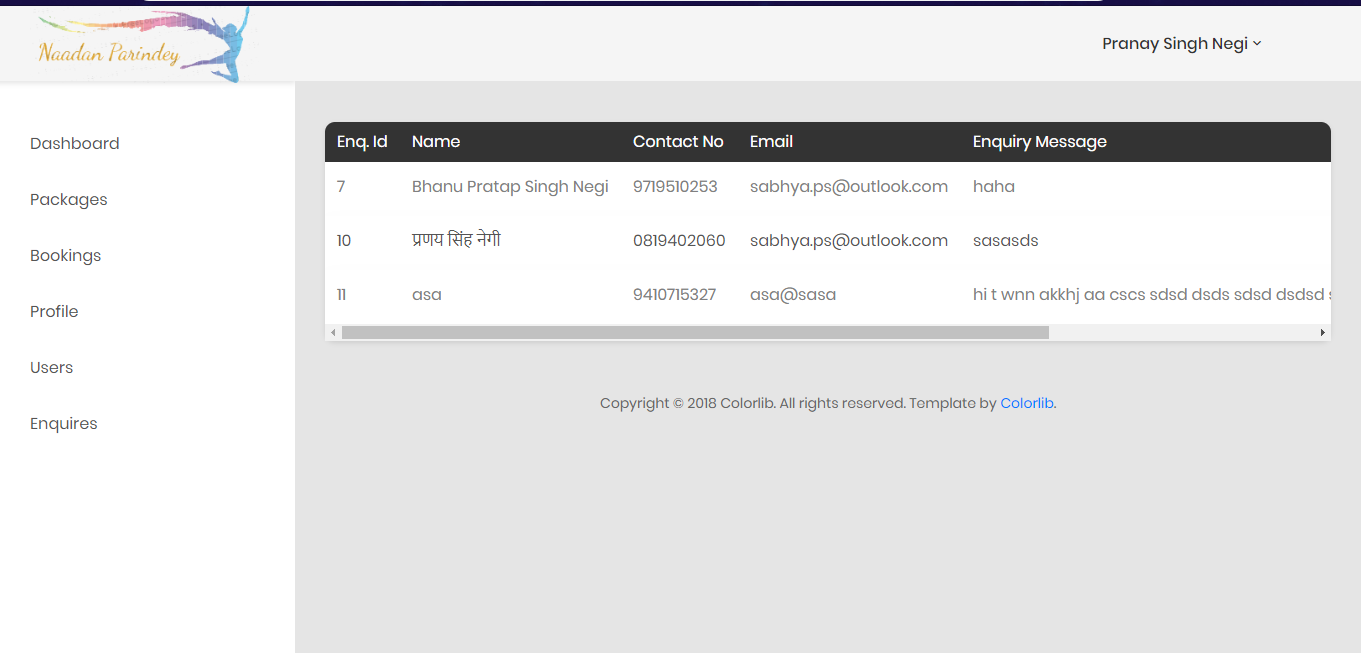


Fig 28: Enquires

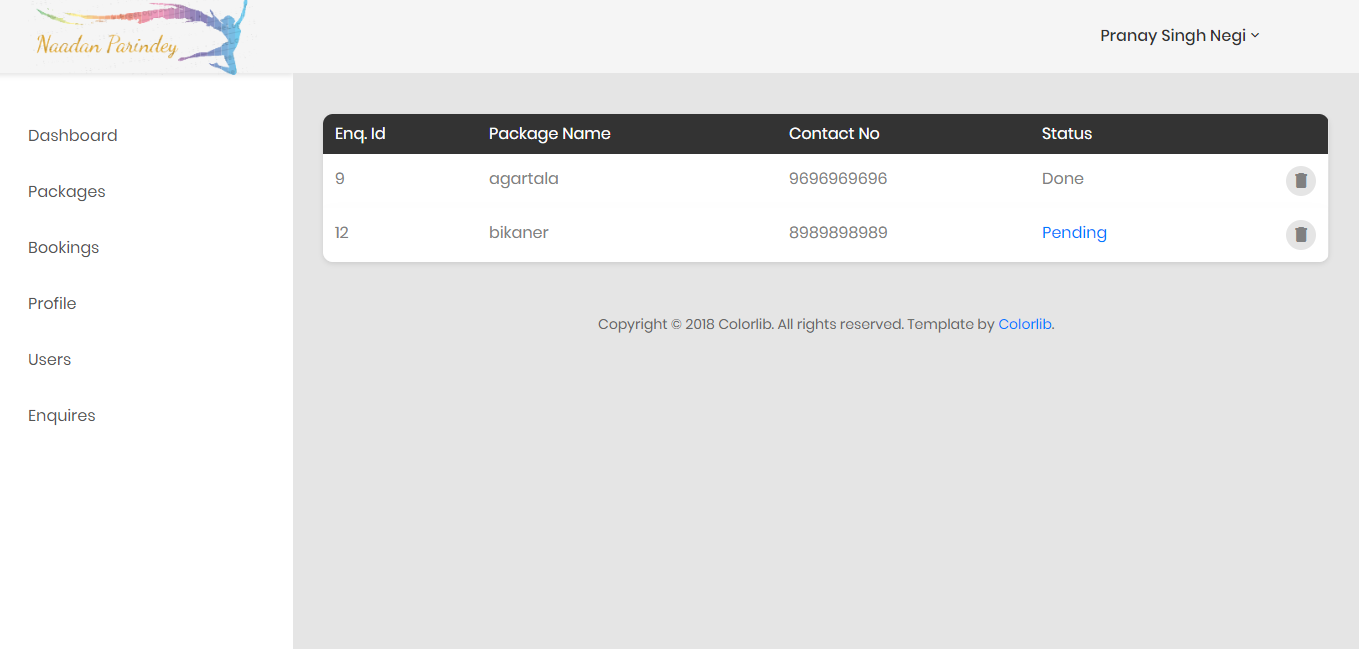


Fig 29: Requests for call-enquiry

1. TESTING

Introduction In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program.

A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended.

Software testing is the technical investigation of the product under test to provide stakeholders with quality related information. Software testing may be viewed as a sub-field of Software Quality Assurance but typically exists independently (and there may be no SQA areas in some companies). In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the code or deliver faster.

* 1. **Definition:**

Software Testing is the process used to help identify the correctness, completeness,security, and quality of developed computer software. Testing is a process of technicalinvestigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it isintended to operate. This includes, but is not limited to, the process of executing aprogram or application with the intent of finding errors. Quality is not an absolute; it isvalue to some person. With that in mind, testing can never completely establish thecorrectness of arbitrary computer software; testing furnishes a criticism or comparisonthat compares the state and behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all businessprocess areas, not just testing.

* 1. **Testing Methodologies:**
     + **Black Box Testing:**

It is the testing process in which tester can perform testing on an application withouthaving any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

* + - **Black Box Testing:**

It is the testing process in which tester can perform testing on an application withhaving internal structural knowledge. Usually The Developers are involved in white box testing.

* + - **Gray Box Testing:**

It the process in which the combination of black box and white box tonics’ areused.

* 1. **Types of Testing:**
     + **Regression Testing:**

It is one of the best and important testing. Regression testing is the process in whichthe functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.

* + - **Re-Testing:**

It is the process in which testing is performed on some functionality which is alreadytested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.

* + - **Static Testing:**

It is the testing, which is performed on an application when it is not been executed. Ex:GUI, Document Testing

* + - **Dynamic Testing:**

It is the testing which is performed on an application when it is being executed.Ex: Functional testing.

* + - **Alpha Testing:**

It is a type of user acceptance testing, which is conducted on an application when it isjust before released to the customer

* + - **Beta-Testing:**

It is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.

* + - **Compatibility Testing:**

It is the testing process in which usually the products are tested on theenvironments with different combinations of databases (application servers, browsers…etc.) In order to check how far the product is compatible with all theseenvironments platform combination.

* + - **Installation Testing:**

It is the process of testing in which the tester try to install or try to deploy the moduleinto the corresponding environment by following the guidelines produced in thedeployment document and check whether the installation is successful or not.

## **Test cases:**

* + - **Login**

|  |
| --- |
| Test case id:t1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case id | Test scenario | Test Steps | Test Data | Expected  Results | Actual results | Pass/fail |
| 1 | Check user login with valid data | Step 1- Run the application, you will be prompted to enter username and password.  Step 2- Enter username  Step 3- Enter Password  Step 4- Click Login Button | Username- prny  Password: qwerty | User should login into application | As  Expected | Pass |
| 2 | Check user login with in-valid data | Step 1- Run the application, you will be prompted to enter username and password.  Step 2- Enter username  Step 3- Enter Password  Step 4-Click Login Button. | Username: prny  Password: 123456 | User should not be able to login into application | As  Expected | Fail |

* + - **Candidate Registration:**

|  |
| --- |
| Test case id:t2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case id | Test scenario | Test Steps | Test Data | Expected  Results | Actual results | Pass/fail |
| 1 | Click directly on Register button without giving any data. | Step 1- Run the application, you will be prompted to enter username and password.  Step 2- Enter username  Step 3- Enter Password  Step 4- Click Register Button | No Data given | Error should be reflected  “Personal details are mandatory to fill” and user should not be able to register. | As  Expected | Pass |
| 2 | Only email id is entered incorrectly and rest of data is given correctly. | Step 1- Run the application, you will be prompted to enter username and password.  Step 2- Enter username  Step 3- Enter Password  Step 4- Click Register Button | Email ID.  Email ID: .nkholiya@gmail.com | Error should be reflected “Either email-id is incorrect or mobile number is not of 10 digits  ” and user should not be able to register. | As  Expected | Pass |

* + - **Add Packages :**

|  |
| --- |
| Test case id:t3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case id | Test scenario | Test Steps | Test Data | Expected  Results | Actual results | Pass/fail |
| 1 | Trying to add the tour packages. | 1. Run the application.  2. Enter username and password and click on submit button.  3.Click Add Packages | User Type- Admin | Package Added Successfully | As  Expected | Pass |

* + - **Booking of Package :**

|  |
| --- |
| Test case id:t4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case id | Test scenario | Test Steps | Test Data | Expected  Results | Actual results | Pass/fail |
| 1 | Trying to book the packages. | 1. Run the application.  2. Enter username and password and click on submit button.  3.Click Booking | Username: ps121  Password: 12345 | Booking Successfully | As  Expected | Pass |

1. CONCLUSION

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.

The proposed system is a web based application and maintains a centralized repository of all related information. The system allows one to easily access the relevant information and make necessary travel arrangements. Users can decide about places they want to visit and make bookings online for travel and accommodation.