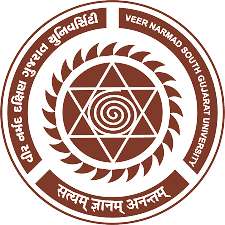
**VIDHYADEEP INSTITUTE OF COMPUTER &**

**INFORMATION TECHNOLOGY, ANITA-KIM**

**VEER NARMAD SOUTH GUJARAT UNIVERSITY**

**(VNSGU)**

**PROJECT REPORT**

**ON**

**“BUDGET SUITES”**

**AS A PARTIAL REQUIREMENT FOR THE DEGREE OF**

**BECHOLOR OF COMPUTER APPLICATION**

**[ B.C.A ]**

**Year :- 2023 – 2024**

**SUBMITTED BY : INTERNAL GUIDE :**

Virag Bavadiya (E21110403000110004) Mrs. Heena Patel

Darshan Mangukiya (E21110403000110060)

Yash Bhimani (E21110403000110011)

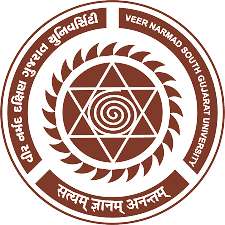
**I/C Principal**

**Mr.**

**VIDHYADEEP INSTITUTE OF COMPUTER &**

**INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the project report, submitted for the project entitled **“Budget Suites”** has been carried out by **“Virag Bavadiya (E21110403000110004)”** at Vidhyadeep Institute of Computer & Information Technology, Kim for partial fulfilment of BCA degree to be awarded by Veer Narmad South Gujarat University for year 2023-2024. This project work has been carried out under my supervision and is to my satisfaction.

Place : ANITA, KIM

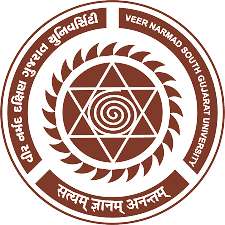
Date :

**Internal Guide I/C Principal**

**VIDHYADEEP INSTITUTE OF COMPUTER &**

**INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the project report, submitted for the project entitled **“Budget Suites”** has been carried out by **“Darshan Mangukiya (E21110403000110060)”** at Vidhyadeep Institute of Computer & Information Technology, Kim for partial fulfilment of BCA degree to be awarded by Veer Narmad South Gujarat University for year 2023-2024. This project work has been carried out under my supervision and is to my satisfaction.

Place : ANITA, KIM

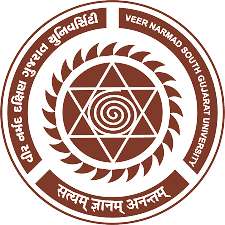
Date :

**Internal Guide I/C Principal**

**VIDHYADEEP INSTITUTE OF COMPUTER &**

**INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the project report, submitted for the project entitled **“Budget Suites”** has been carried out by **“Yash Bhimani (E21110403000110011)”** at Vidhyadeep Institute of Computer & Information Technology, Kim for partial fulfilment of BCA degree to be awarded by Veer Narmad South Gujarat University for year 2023-2024. This project work has been carried out under my supervision and is to my satisfaction.

Place : ANITA, KIM

Date :

**Internal Guide I/C Principal**

**Project Title Report**

To,

Vidhyadeep Institute of Computer &

Information Technology,

Anita-Kim.

Virag bavadiya, Darshan Mangukiya, Yash Bhimani students of Vidhyadeep Institute of Computer & Information Technology have completed their project work on ‘Budget Suites’ under the guidance of Mrs.Heena Patel during the academic semester from January 2024 to April 2024.

**Internal Guide**

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr. No | Topic Name | Page No. |
| 1 | Introduction | 1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**1.Introduction**

**1.1 Project Profile**



|  |  |
| --- | --- |
| Project Title | **Budget Suites** |
| Project Type | **Web Application** |
| Front End | **React Js.** |
| Back End | **Nest Js.** |
| Duration | **3 Months** |
| Internal Guide | **Mrs.Heena Patel** |
| Submitted To | **Vidhyadeep Institute of Computer & Information Teachnology** |
| Submitted By | **Bavadiya Virag B. [6465]**  **Mangukiya Darshan V. [6512]**  **Bhimani Yash P. [6472]** |

**1.2 Project Overview**

Budget Suites is booking services for hotels. The main objective of the project is to create a Budget Suites that allows users to search and book a hotel online based on country, location, room, room types etc.

The selected hotels are displayed in a tabular format and the user can book their online reservation through credit card payment. Using this website the user can book a room instead of going out toa hotel and wasting time.

There are many Budget Suites like bookings, Untamed which were designed using Html. I want to develop similar website using Nest Js, MongoDB.

Budget Suites is an online web application where the customer can book hotel online.

Through a web browser the customers can search a hotel by its country, location, room and room type can add to the favourites and finally book through online payment.

The user can login using his account detail or new user can set up an account very quickly. They should give the details of their name, contact number, email id.

Ther user can also give feedback to a hotel by giving rating on a score of five. The hotels are divided into many categories based on subject like Software, Database, English And Architecture etc.

The Budget Suites website provides users with online booking through a web browser. A user can create, sign in to account, add hotels into a favourites list and booking by online payment methods.

The administrator will have additional functionalities when compared to the common user. He can add, delete and update hotel details, room type categories, user information and also confirm booking and also maintain services.

This application is developed using React Js, Nest Js programming language. The data sets, data grids, user control are used to develop the Budget Suites.

**2. System Environment**

**2.1 Project Description**

**Memory 1GB**

**Hard Disk 10GB**

**2.2 Software Requirements**

**Back End MongoDB Server 1.42.1**

**Front End Google Chrome**

**O.S. Windows 10**

**Software Visual Studio Code Editor 2010**

**3. Problem Specification**

**3.1 Introduction, Objective, Purpose**

* It is Difficult to manage E-commerce site so we created the website that can be easily managed by admin and user.
* It is complicated to manage all functionalities so here admin can able to view all function easily like searching, booking etc..
* Using Budget Suites the user can see and also book hotel online.
* User can apply booking Budget Suites and book hotel from website.

**3.2 System Model**

In Development of software the working model plays important role that is “prototype working model”. Prototype is considered as a development methodology. It is an engineering inspired approach to system development. In which development life cycle the working model of the proposed system is created, tested and reworked until a finally prototype working model is available and by using finally prototype working model one can develop a new system.

**3.3 Prototype Model**

In this prototype model before designing phase, a prototype is developed, tested, reviewed and approved by the customer, after that design will be ready for coding, testing, installation and maintenance will take place. This prototype is prepared based on the user requirements. Prototype testing checking for the required components are present or not. By using this prototype, customer can understand the requirement of desired system and also the user can get an “actual feel” of the system. It is an attractive idea for complex and bigger system.

**3.4 Feasibility Study**

This system is feasible to adapt to any E-commerce website for Booking. It will take some resources for running the system. User can book the hotel.

Feasibility study included following terms

**3.4.1 Behaviour Feasibility :**

Budget Suites user-friendly interface for view the particular hotels and also provide the facility to view hotel. This website runs on maximum configuration.

**3.4.2 Technical Feasibility :**

The technical feasibility assessment is focused on gaining and understanding of the present technical resources of the E-commerce system and their applicability to the expected needs of the system.

**3.4.3 Economical Feasibility :**

In this project, we will require to have an internet connection for better online website. Such at the developer end it needs the good server with high capacity of RAM and CPU processor so can it can handle lots of members at time online communications but as it will be web-enabled we do not have any extra cost of setting up a network.

**4. Risk Identification and Management**

**4.1 Risk Monitoring**

Risk :- Developer **Inexperienced**

With the development of the software, we have defined some Proactive risks before technical work initiated. This part of document includes the risk management step.

|  |  |
| --- | --- |
| **Risk :** | **Unrealistic Deadline** |
| **Probability :** | **70%** |
| **Description :** | **Because of the short duration and large application there is probability not to complete the project within deadline.** |
| **Contingency Plan:** | **From the starting day of the project, we have worked hard and very speedily to complete the project within deadline.** |

|  |  |
| --- | --- |
| **Risk :** | **Customer will change requirements** |
| **Probability :** | **60%** |
| **Description :** | **The customer will change the requirements very frequently as the project begins. So, we have probability not to complete the project as the customer required.** |
| **Contingency Plan:** | **From the past project, we have knowledge about it. So we interact with customer every time new work begins and conduct FTR & revise the work as the customer requires.** |

**Technical Risk**

* **Language Difficulties**

The Familiarity with the front-end language is illustrated in this risk.

* **The reasons behind this risk are as follows :**

The project is mandatory built on React Js. Platform. Since this language is new

Programmers, the programmers might face some difficulties.

As no one from the programmer is partially familiar with this platform, some functionalities might not considered in implementation or some functionalities might not achieved completely.

* **Specification ambiguity**

The complexity of the requirements or inappropriate understanding is included in this risk.

* **The reasons behind this risk are as follows :**

The specification of this projects little profound and hard to identify.

* **Technology Unknown**

The risk is concerned with the unrealised technical conditions**.**

* **The reasons behind this risk are as follows :**

The platform we should be working on is client-server environment, which is introduction to all of us.

The lack of conceptual implement on this kind of environment is new.

**5. Technology Used for Development**

**Front End :** React JS with Nest JS

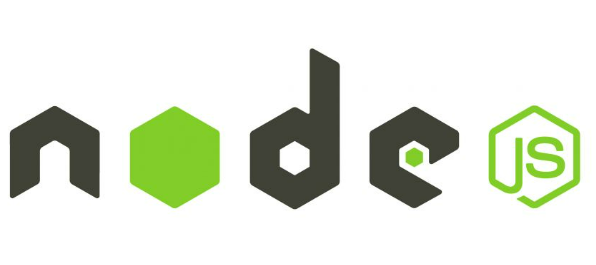
**Back End :** MongoDB 1.41.2

**Documentation Tools :** Microsoft Office Word 2021

**Hardware :** Hard disk 500GB, RAM 2 GB

**Node.js**

* Node.js is a cross-platform, open-source to write command line tools and for server-side scripting. The ability to run JavaScript code on the server is often used to generate dynamic web page content before the page is sent to the user’s web browser.



* Node.js is an event driven architecture capable of asynchronous I/O. These design choices aim to optimize throughout and scalability in web application with many input/output operations as well as for real-time Web applications.

**NPM (Node Package Manager)**

* NPM is a package manager for the JavaScript programming language maintained by NPM,

NPM is the default package manager for the JavaScript runtime environment Node.js and is included as a recommended features in the Node.js installer.

* It consists of a command line client, also called NPM, and an online database of public and paid-for private packages, called the NPM registry. The registry is accessed via the client, and the available packages can be browsed and searched via the NPM website. The package manager and the registry are managed by NPM.



**React JS**

* React JS is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta (Facebook) and a community of individual developers and companies.



**React**

* React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js Because React is only concerned with the user interface and rendering components to the DOM, React applications often rely on libraries for routing and other client-side functionality.
* A key advantage of React is that it only renders those parts of the page that have changed, avoiding unnecessary rendering of changed DOM element.

**Nest JS**

* Nest JS is a framework for building efficient, scalable Node.js server-side application.it uses progressive JavaScript, is built with and fully supports TypeScript and combines of OOP (Object Oriented Programming), FP (Functional Programming) and FRP (Functional Reactive Programming).



* Nest makes use of robust HTTP server framework like Express and optionally can be configured to use Fastify as well.
* Nest provides a level of abstraction above these common Node.js frameworks, but also exposes their APIs directly to the developer.
* This gives developers the freedom to use of third-party modules which are available for the underlying platform.

**Ajax**

AJAX = Asynchronous JavaScript and XML. AJAX is not a programming language. AJAX just uses a combination of :

* A browser built-in XMLHttpRequest object (to request data from a web server).
* JavaScript and HTML DOM (to display or use the data)

Browser Server

* Process HttpRequest
* Create a response and send data back to the browser
* Process the returned data using JavaScript
* Update page content

An event occurs…

* Creates an XMLHttpRequest object
* Send HttpRequest

**Internet**

Browser

**Internet**

**Java Script**

**JavaScript**

JavaScript is a very powerful **client-side scripting language.** JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used in game development and **Mobile** application development.

**jQuery**

jQuery is a lightweight, ”write less, do more”, and JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.



jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

* HTML / DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities

**Why jQuery?**

There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

* Google
* Microsoft
* IBM

**MongoDB Collection**

* MongoDB is a source-available, cross-platform, document-oriented database. Classified as a NoSQL database product, MongoDB utilizes JSON-like document with optional schemas. MongoDB is developed by MongoDB Inc., and current versions are license (SSPL). It’s 5th most popular database engine.



* MongoDB supports field, range query and regular-expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of result of a given size.

Some Typical database administration and programming tasks could include:

* + Create & maintain databases
  + Create & maintain tables
  + Create & maintain other database objects such as stored as stored procedures, views, etc.
  + Create & maintain and schedule data backups
  + Replication (e.g., create a copy of the database)
  + Create & maintain users, roles, etc
  + Optimization tasks

**6. Planning**

**6.1 System planning**

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile for tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model:

Testing

Planning

Building

Designing

Building

Req. Analysis

Testing

Testing

Planning

Building

Designing

Planning

Req. Analysis

**2-3 months**

-

**2-3 months**

Designing

Req. Analysis

**6.2 Time-line Chart**

Once a project is found to be feasible, we undertook project planning. Project planning is undertaken and completed even before any development activity starts. The objective of project planning is to provide a framework that enables us to make reasonable estimates of resources, cost and schedule.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Week 1** | | | | | | | | **Week 2** | | | | | | | **Week 3** | | | | | | | **Week 4** | | | | | | | |
| Learn React JS and Nest JS |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perform CRUD in React JS |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirement Gathering |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Background study |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Database Creation |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Identification of End-Users |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Use case Diagrams |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design Rough GUI |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UML Diagrams |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Admin side design start |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | **Week 5** | | | | | | | | **Week 6** | | | | | | | **Week 7** | | | | | | | **Week 8** | | | | | | | |
| Back end start |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Basic functionalities implementation |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prepare PPT & Partial Report |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing of basic functionalities |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementation of Changes |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Remaining functionalities |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Language, Skill |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Admin side complete |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | **Week 9** | | | | | | | **Week 10** | | | | | | | **Week 11** | | | | | | | **Week 12** | | | | | | |
| Fixing bugs & Errors |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prepare Final Report |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**7. System Analysis**

**7.1 Database Design**

|  |
| --- |
| **amenities** |
| \_id |
| amenities\_name |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |

|  |
| --- |
| **bookings** |
| \_id |
| check\_in |
| check\_out |
| user\_id |
| hotel\_id |
| location\_id |
| country\_id |
| total\_rooms |
| total\_children |
| room\_details |
| user\_details |
| total\_amount |
| transaction\_id |
| payment\_method |
| payment\_status |
| booking\_status |
| checkout\_url |
| created\_at |
| updated\_at |

|  |
| --- |
| **countries** |
| \_id |
| country\_code |
| country\_name |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |

|  |
| --- |
| **hotels** |
| \_id |
| location\_id |
| country\_id |
| hotel\_name |
| hotel\_code |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |
| hotel\_address |
| total\_rooms |
| hotel\_image |
| roomTypes |
| amenities |

|  |
| --- |
| **locations** |
| \_id |
| country\_id |
| location\_code |
| location\_name |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |
| location\_image |

|  |
| --- |
| **offers** |
| \_id |
| offer\_type |
| offer\_code |
| offer\_amount |
| isOneTime |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |
| expired\_on |

|  |
| --- |
| **roomtypes** |
| \_id |
| roomtype\_name |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |
| max\_adults |
| max\_children |

|  |
| --- |
| **users** |
| \_id |
| fullname |
| email |
| password |
| phone |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |
| profileImg |
| userType |

|  |
| --- |
| **Verificationcodes** |
| \_id |
| userId |
| generationTime |
| verificationCode |

|  |
| --- |
| **rooms** |
| \_id |
| hotel\_id |
| room\_type\_id |
| price |
| total\_rooms |
| createdAt |
| updatedAt |
| isDeleted |
| isActive |

**7.2 E-R Diagram**

A database can be modelled as: “a collection of entities, relationship among entities.” An entity is an object that exist and is distinguishable from other objects. “Example: specific person, company, event, plant! Entities have attributes” Example: people have names and addresses! An entity set is a set of entities of the same type that share the same properties. “Example: set of all person, companies, tress, holidays.

A data model is a conceptual representation of the data structure that are required by a database. The first step in designing a database is to develop an Entity-Relation Diagram (ERD). The ERD serves as a blue print from which a relational database maybe deduced.

The ERD for the project and later we will show the transformation from ERD to the relational model.

Rectangle represents entity sets! Diamond represent relationship sets! Lines link attributes to entity sets and entity sets to relationship sets! Ellipses represent attributes! Double ellipses represent multivalve attributes! Dashed ellipses denote derived attributes! Underline indicates primary key attributes.

**7.3 Data Flow Diagram**

Data flow diagram shoe the flow of data from external entities into the system, and from one process to another within the system.

There are four symbols for drawing a DFD:

1. Rectangle representing external entities, which are sources or destinations of data.
2. Ellipses representing processes, which take data as input, validate and process it and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items.
4. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stakes of paper. The Data Flow Diagrams for the current system. Each process within the system as a Context Level DFD and later as a Detailed DFD.

The Context Level DFD provides a conceptual view of the process and it’s surrounding input, output and data stores. The Detailed provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

**Context Level Diagram**

**User**

**Admin**

Online Hotel Booking

Hotel

User

Login

User

Booking

**First Level Data Flow Diagram**

**Search Hotel**

**1.0**

**User**

**Book**

**2.0**

**User**

**Search**

**1.1**

**Apply Filter**

**1.2**

**Book**

**Author**

**Category**

**Language**

**View Hotel**

**1.3**

**Book**

**Specification**

**Image**

User\_Detail

UserID Password

**User**

**Admin**

UserID Password

**User**

**Admin**

**Hotel**

HotelID

Hotel Name

Book Hotel Price

Booking

Payment

Confirm Place

UserID BookID

**Second Level DFD**

**Hotel Information**

**Hotel**

**Hotel Name**

**Hotel Name Review**

**Country Location Price**

**Rooms**

**Hotel**

**Country Location Price**

**Hotel Name**

**Hotel Information**

**Hotel Information**

**Hotel Review**

**Hotel**

**Country Location Price**

**Hotel Name Review**

**Hotel Name**

**User**

**Hotel Name Review**

**Hotel Name Review**

**Hotel**

**Rooms**

**Hotel Information**

**Change Password Second Level DFD**

**UserID**

**Password**

**UserID**

**UserID**

**Password**

**UserID**

**Password**

**UserID**

**Password**

**UserID**

**Password**

**UserProfile**

**Login**

**User**

**Customer Authentication Second Level DFD**

**UserId HotelID Price**

**History**

**HotelName UserID**

**Price**

**UserID**

**Password**

**HotelID Price**

**UserID Password**

**UserID**

**UserID Password**

**Hotel**

**Book History**

**Authentication User**

**Authentication User**

**User**

**Hotel**

**Add to Wishlist**

**HotelName UserID**

**Price**

**Customer Authentication User Profile DFD**

**UserID**

**Password**

**UserID**

**Password**

**User**

**Authentication User**

**User Booking Second Level DFD**

**Add Rooms**

**HotelID**

**Hotel**

**User**

**HotelID Price**

**Rooms**

**UserID**

**Password**

**11. References:**

**HotelID Payment\_type**

**Price**

**Payment\_type**

**Price**

**UserID HotelName**

**Price**

**Booking**

**User Login**

**Payment**

**PaymentType**