ONLINE HOTEL BOOKING

A Project Report for Industrial Training and Internship

submitted by

DEEP BISWAS

In the partial fulfillment of the award of the degree of

B.Tech

in the

CSE

Of

TECHNO ENGINEERING COLLEGE BANIPUR

At

Ardent Computech Pvt. Ltd.

CERTIFICATE FROM SUPERVISOR

This is to certify that **"Deep Biswas, 24400123018**" have completed the project titled "Online Hotel Booking" under my supervision during the period from "<19/08/2026>" to "<14/10/2026>" which is in partial fulfillment of requirements for the award of the **B.Tech** degree and submitted to the Department of "**CSE**" of **"Techno Engineering College Banipur"**.

_____

Signature of the Supervisor

Date: dd/mm/yy

MR. Anindya Mukherjee
Name of the Project Supervisor:

BONAFIDE CERTIFICATE

Certified that this project work was carried out under my supervision

"Online Hotel booking" is the bonafide work of

Name of the student : Deep Biswas ***Signature: Deep Biswas***

SIGNATURE

Name : Anindya Mukherjee

PROJECT MENTOR

SIGNATURE

Name:

EXAMINERS

Ardent Original Seal

ONLINE HOTELS AND RESORTS BOOKING SYSTEM

J2EE PROJECT

Submitted by:

DEEP BISWAS
24400123018
CSE
TECHNO ENGINEERING COLLEGE BANIPUR

Under the guidance of Anindya Mukherjee on behalf of

ARDENT COMPUTECH PVT. LTD

CONTENTS

SERIAL NO	TOPICS	PAGE NO
1	Abstract	4
2	Introduction	5
3	Problems of current system	6
4	Concept & Problem Analysis	7-11
5	Web Application Survey	12
6	SDLC of the project	13-18
7	Database Design	18-19
8	Use Case Diagram	20
9	Entity Relationship Diagram	21
10	Data Flow Diagram	22
11	Development	23-52
12	Physical Design	53-71
13	Conclusion	72
14	Limitations	73
15	Future Scope of Project	74
16	Bibliography	75

ABSTRACT

Hotel reservation system is an important part in the life of a modern hotel, because it ensures proper work of the hotel, making it efficient and provides the option to book a room online. It holds information for the workers and administration about the rooms and the hotel overall. This system removes most of the paperwork making it an obligatory asset to have for every modern hotel. Analyzes and improves the current reservation system. The aim of this work is to provide in-depth analysis about the system. Search of theory, regarding hotels, reservations, online reservations and online reservation systems.

Attempt to improve the current situation in hotel reservation business. Also to create a small prototype, showing how the system will work from the user side.

INTRODUCTION

Online hotel and resort booking website which provides the facilities for booking hotels and resorts for customers. They operate their business in all the cities of India as well as the world. Any customer who need to book hotel or resort had to visit their office which is situated in India for checking the availability as well as negotiation. Recently they have decided to change their booking pattern from manual to web based system. It will help them to manage the customers booking easily and also to keep the customers data more safe. It will also help staffs to keep in track their customer's online booking request as well as easily to reply feedback to the customers. For that they hired a developer to build the web system for this reputed company. In this documentation it shows how the web based system will be implemented and looks like (User Interface). Since the web application is available in the Smartphone it is easily accessible and always available.

PROBLEMS OF CURRENT SYSTEM

As ***Let's Go*** is having manual booking system, they are facing some problems issuing booking requests of customers. All the necessary booking stuffs are being done in hard copy. So it becomes much difficult for staffs to keep the records updated all the time. As for example, if the customers need to change the check in date it become difficult for them to find out the customers booking details for updating as there are so many customers booking records. Again, regarding current system customers cannot give feedback online and also staff cannot reply to them promptly. Besides tourists from other countries need to call directly for booking purposes. So they cannot get the chance to view their apartment rooms or hotels rooms before they make book.

Secondly, in hotels one needs to use the reception desk to check-in and get your hotel number. That can have good and bad parts. For example, a bus with tourists arrives to a hotel at night and only one receptionist is working. He has to do everything manually and it can take an extremely long time. Guests are tired, but they have to fill out some forms or just wait in line for over an hour to get their rooms sorted, while with a QR code it's possible to receive your room card in a bit over a minute.

For a guest to check in he has to wait for the receptionist to be available, then the worker asks for reservation information. Often that information can be printed out on paper or just documents are used to prove a reservation. Receptionist has to provide guest forms to be filled and/or just make the client wait, sometimes longer than it should be, because he has to find the reservation. Check room availability, if it's clean and ready to be used, confirm and check every little detail possible. It can take a very long time and if there are a lot of guests waiting for service it can get extremely frustrating for the guest. Other problem, are days when you don't want to deal with people. For example, you are very tired and just want to get to your room and relax after a long flight/trip/train ride. Guest and receptionist can be both in a bad mood and then problems can occur.

CONCEPT AND PROBLEM ANALYSIS

It is a process of understanding the real-world problems and user's needs and proposing abstract solutions to those problems. It can be divided into 5 sub tasks:

- **Cost Analysis**
- **Time Analysis**
- **Team Structure**
- **Software Configuration Management**
- **Quality Assurance Plan**
- **Risk Management**

1. Cost Analysis :

In this project, we have applied the method of **Function Point Analysis**. It is an advanced technique of determining the size of a project where the size depends on the functionalities of the project. Function Point helps us in determining the exact size of the project by considering various value adjustment factors and basic 5 parameters which are very crucial with respect to any software application. There are 14 value adjustment factors which are considered on a scale of 0-5. The basic 5 parameters which are also considered are, External inputs, External outputs, External inquiries, Logical internal files, Number of interfaces.

Function point is computed in three steps.

- The first step is to compute the unadjusted function point (UFP).
- In the next step the UFP refined to reflect the differences in the complexities of the different parameters of the expression for UFP computation.
- In third and final step, Function Point is computed by future refining UFP account for the specific characteristics of the project that can influence the development effort.

UFP = (Number of input) * 4 + (Number of outputs) * 5 + (Number of queries) * 4 + (Number of logical files) * 10 + (Number of interfaces) * 7

Finally Function Point (FP) is given as the product of UFP and TCF. That is, FP = UFP*TCF

Where, TCF(Technical Complexity Factor)=0.65+(0.01*Σf)

2. Time Analysis :

Time analysis is a specification which gives a description of the time period required to complete the different tasks. The project task scheduling is taken up in order to decide which taken up in order to decide which tasks would be done when inside the schedule.

In our project the different analysis is shown as :-

TimeLine Chart

Development Phase	120 Days						Duration (Day)
	0 to 15 Day	16 to 30 Day	31 to 45 Day	46 to 60 Day	61 to 75 Day	76 to 90 Day	
Requirement Gathering							10
Analysis							15
Design							30
Coding							25
Testing							12
Implementation							08
Documentation							80
Total Time (Days)							90

Task Time Analysis

4. Software Configuration Specification :

Changes continuously take place in a software project. Changes due to evolution of work products. As the project proceeds changes due to bugs being found and fixed and changes due to requirement modifications. Software Configuration Management(SCM) are the practices and procedures for administering the source code, producing software development builds, controlling change and managing software configuration. Specifically, SCM ensures the integrity, reliability and reproducibility of developing software products from conception to release. **The goal of SCM that are followed during**

Building the project:

- **Configuration Identification:** The objective of this system is to avoid the introduction of errors related to lack of testing as well as incompatibilities with other CIs.
- **Configuration Control:** Implementing a controlled change process. In the project, our mentor reviewed the changes in our project.
- **Configuration Status Accounting:** Recording and reporting all the necessary information on the status of the development process to the project guide.
- **Configuration Auditing:** Ensuring that configuration contain all their intended parts and are sound with respect to their specifying documents, including requirements and architectural specification.
- **Build Management:** Managing the process and tools used for builds. One of the most important steps of a software build is the compilations process.
- **Process Management:** Process Management is the application of knowledge, skills, tools, techniques and systems to define, control, report and improves the process with the goal to meet the requirements profitably.
- **Environment Management:** Managing the software and hardware that host the system.

5.Quality Assurance Plan :

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality.

The main purpose of the Software Quality:

Quality is a deciding factor of a product. If the quality is compromised then the value and need of the product is also reduced. So it is an earnest effort of our team to develop the website **"Online Hotels & Resorts Booking System"** which can be reliable, user friendly and secured. The Quality Assurance (QA) Plan involves defining or selecting standards that should be applied to the software development process or software product. There are two types of standard that may be established as part of the Quality Assurance Plan:

- **Product standards:** These are standards that apply to the software product being developed.They include document standards such as the structure of the requirements document which should be produced, documentation standards such as standard comment header for an object class definition and coding standards which define how a programming language should be used.
- **Process standards:** These are standards that define the processes which should be followed during software development. They may include definitions of specification, design and validation processes and a description of the documents which must be generated in the course of these processes.

Online hotel and resorts booking Site includes the following software quality assurance:

- **Availability** : Once the website is launched it is available in World Wide Web provided the web server and web hosting is up to the required features and can be accessed with ease.
- **Instability**: The above system does not required any installation.
- **Maintainability**: For the updates and maintenance of the system the source codes of the system are well documented and is hassle free.
- **Accurateness**: The system provides the exact information that is based on product details.
- **Testability**: The system will not be available if some errors occur while easier debugging.
- **Usability**: The system is well designed for easy usage and navigation.The system has good GUI interfaces with buttons for easier understanding for user.

6. Risk Management :

The risk management depends upon three major factors:

- **Time:** Time is a crucial factor that comes in way of the process. In our project we are given 4months to complete our project. But the main problem is that we have taken 2 months to gather all knowledge about the platforms, install them and operate them. Rest of the month we have taken for coding, designing the forms and binding the project.
- **Knowledge:** All members of our group started to work on the project without proper knowledge on it. This turns out to be a serious problem in the long run of the project. The platform we have chosen to develop the project is-**JAVA ECLIPSE IDE**. Now the main fact is that we **don't have enough knowledge on these software or platform which we will be working on. In** that case, it is most important to gather knowledge how to operate this software. We have taken minimum 1 months to gather knowledge on these software.
- **Resource:** As J2EE was not in our syllabus, so it was a problem for us to gather knowledge in time and to complete the project. So we did not receive the expected resource from the organization regarding the subject.

WEB APPLICATION SURVEY

We have visited some particular web sites for our design and implementation purpose. These websites are given below.

https://www.trivago.in/

I followed this site to get some idea how to create the website.



SDLC OF THE PROJECT

This project is a Web Application and going to be developed using J2EE technology like JSP and JDBC. Similar to all other software projects, this project too has its own Software Development Life Cycle (SDLC). The SDLC of this project will be discussed under the following heads.

1. FEASIBILITY STUDY:

The purpose of the feasibility study is not to solve the problem, but to determine the problem is worth solving. This helps to decide whether to proceed with the problem or not. It involves the analysis of the problem & collection of all relevant information relating to the product such as items that would be input to the system, processing required to carried those data, the output data required to be produced by the system as well the various constraints on the behavior of the system. "Web based placement management system" had undergone the feasibility study so that the proposed system is possible for development deployment in our college. The feasibility study concentrates on the following, such as Operational Feasibility, Technical Feasibility, and Economic Feasibility

1.1 ECONOMIC FEASIBILITY: The economic feasibility study evaluate the cost software development against the ultimate income or benefits get from the developed system. There must be scope for profit after the success completion of the project.

1.2. TECHNICAL FEASIBILITY*:* Technical feasibility study compares the level of technology available in the software development firm and the level of technology required for the development of the product. The level of technology consists of the programming language, the hardware resources, other software tools etc.

1.3. OPERATIONAL FEASIBILITY: Operational feasibility study tests the operational scope of the software to be developed. The proposed software must have highoperational feasibility. The usability will be high.

2. TOOLS AND PLATFORM USED FOR DEVELOPMENT :

TECHNOLOGIES :	JAVA 2 EE (JSP, SERVLETS, JDBC)
WEB DEVELOPMENT TOOLS :	HTML, CSS, JAVASCRIPT, XML
WEB SERVER :	TOMCAT 10.1
RDBMS / BACK-END :	ORACLE 10g
IDE :	ECLIPSE
OPERATING SYSTEM :	WINDOWS 10
OTHER APPLICATION SOFTWARES USED :	MS Word (For Documentation)
HARDWARE CONFIGURATION OF THE MACHINE USED FOR DEVELOPMENT :	Intel(R) Core(TM)i3 CPU @2.80GHz , 4GB RAM, 500GB HDD.

3. REQUIREMENT ANALYSIS :

3.1. SOFTWARE REQUIREMENTS:

- **OS**: Windows
- **Database**: Oracle
- **IDE**: Eclipse
- **Server**: Tomcat Server (Free-Open source)
- **Language**: Java-JEE environment (Free-Open source)
- **Front end**: Browser with support for JavaScript

3.2. HARDWARE REQUIREMENTS :

- Processor – i3
- Hard Disk – 50 GB
- Memory – 2GB RAM
- Mouse – Any Standard
- Keyboard – Any Standard
- Monitor – Any color monitor
- Local Area Network Preferable

3.3. FUNCTIONAL REQUIREMENTS :

- The system supports customers booking and able to modify them.
- Customers can search based on hotels,resorts and places (ex. Delhi, Dubai)
- When a customer search for hotels and resorts,the search result must contain hotel or resort information (Address, Ratings, and Price) and also its availability within choosing check in and check out date.
- Customers able to cancel/remove their booking from their account.
- Staffs are able to edit customers booking information (updating check in, check out, room preferences, bed preferences and also cancelling booking).

- Customers can book online and pay with credit or debit card.
- The system must send booking confirmation email after successful payment.
- Customers can write reviews about hotels and apartment and also rate them.
- Customers able to check their booking status from their individual account.
- Customers can send feedback or call the company for booking purposes.
- Customers can check for latest promotion or deal.

3.4. NON-FUNCTIONAL REQUIREMENTS :

- The system must ensure that all the transferable data as for examples customers credit or debit card number, CVV Code, e-payment should be done in secured connection.
- The system must be able to handle multiple transactions a time.
- The system must provide customers 24*7 hours online booking service.
- The system should support almost all the browsers (Internet Explorer, Safari, Chrome, and Firefox).
- System should send the newsletter about ongoing promotions or deal to registered customers.

3.4. EXTERNAL INTERFACE REQUIREMENTS :

Admin Interface

Some of the user interface screens are described in the following table :

Screen Name	Description
Login	Only admin can log in to the site.

Table 4:External Interface Requirements

Error Messages

Yet the system has been tested properly on the local host machines and also the system generated errors have been checked by using exception handling mechanism but there can be also some error occur at the runtime unexpectedly due to server failure.

Here we have also provided some user defined error messages which act as a warning message to user. This user defined error messages do not follow any number to identify them. We have used a particular warning message for particular case when a user does not properly interact with them. For example, if user properly log-in then **'Login successful'** alert message is shown**,** otherwise, **'Incorrect email-id or password!!!'** is visible.

4. REQUIREMENT DOCUMENTATION:

This step involves documenting the requirements in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. A requirement specification document is categorized in different ways according to the stakeholders' need, helping to create a clear contract between development and business. The following sections include the different segments –

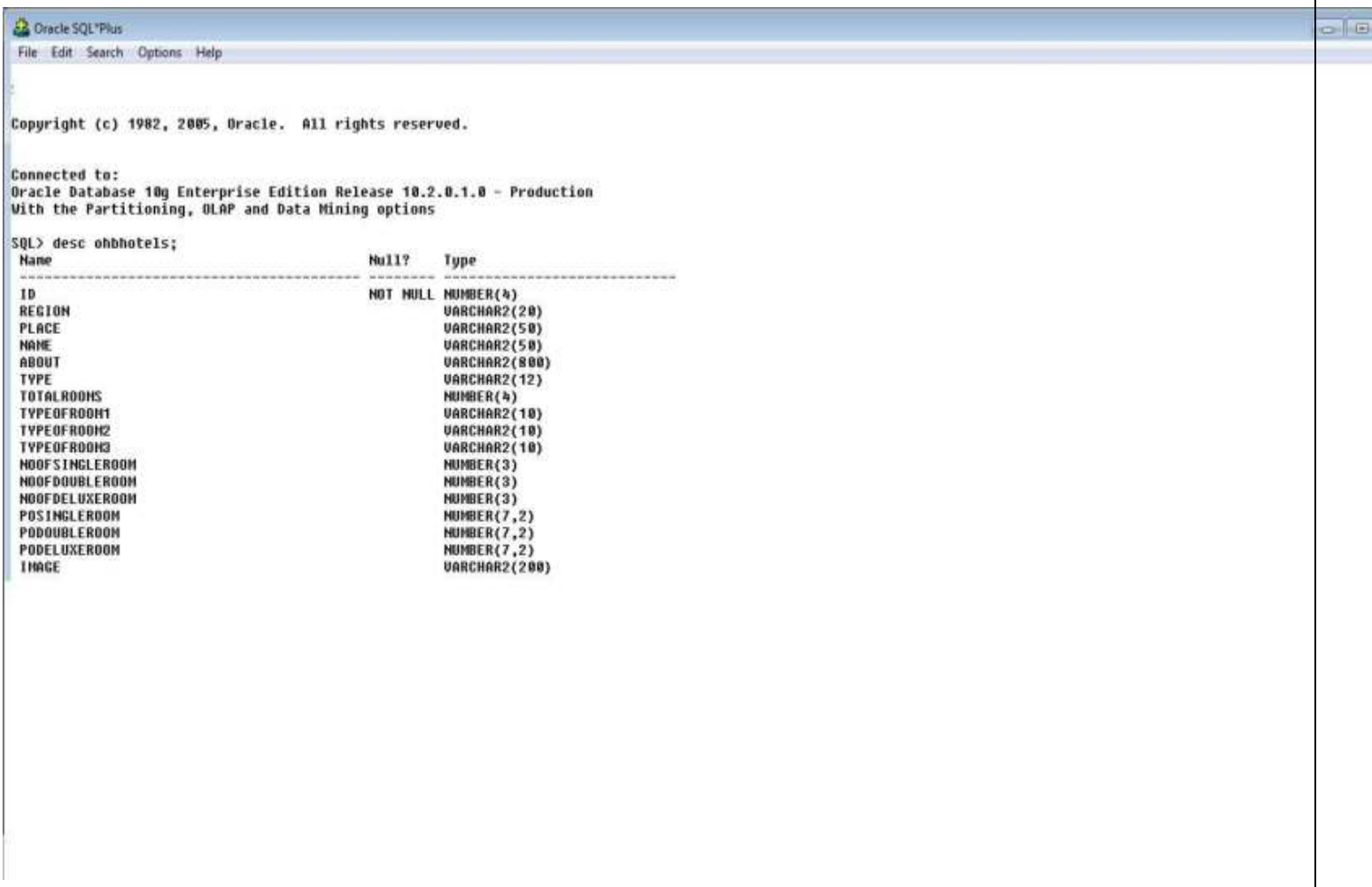
This document is for the developers, testers and customers to make the understanding of the project is easier. The following explanation of the abbreviations will help in the following purpose.

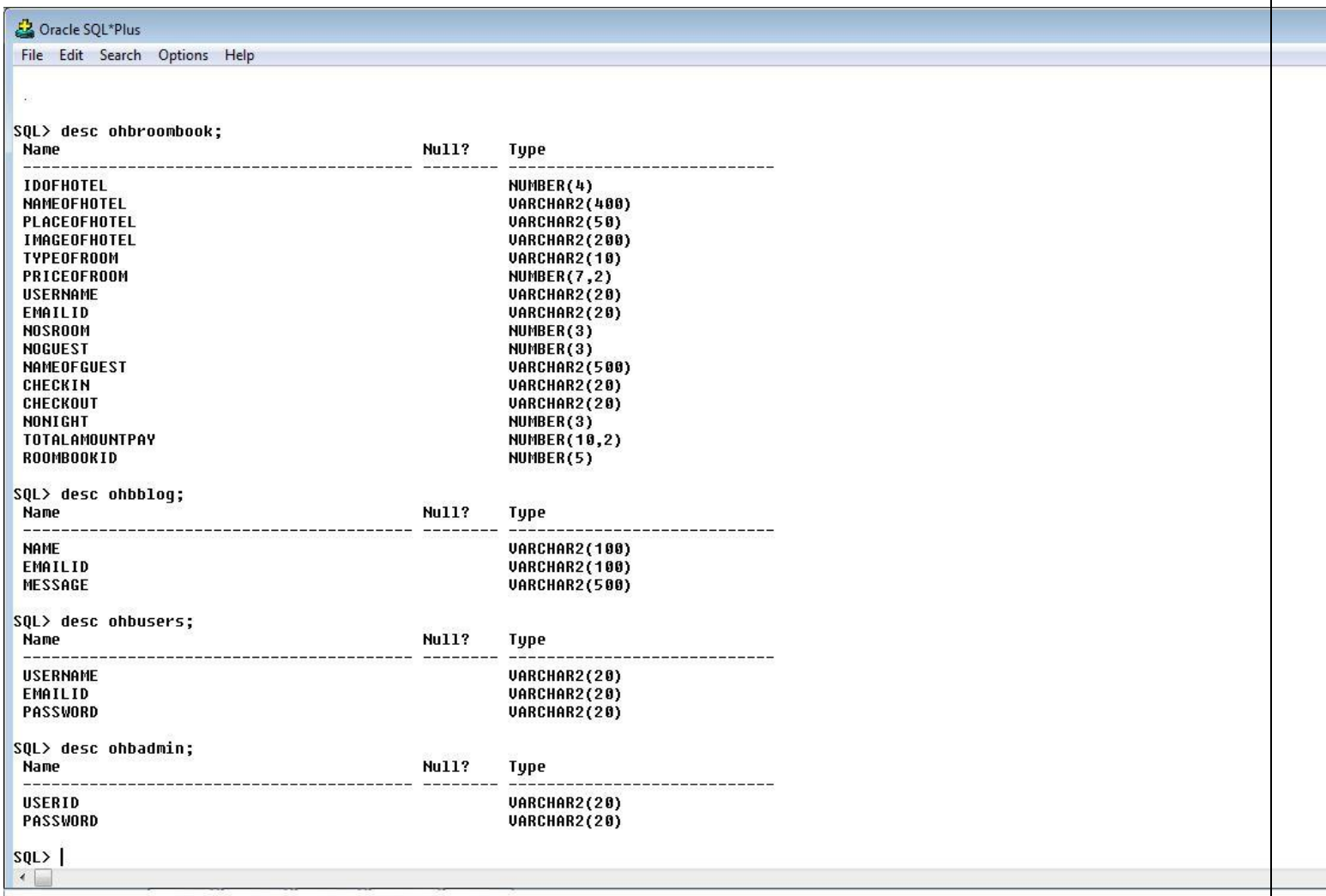
ABBREVIATIONS	FULL FORM
J2EE	Java Enterprise Edition
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JSP	Java Server Pages
Admin	Administrator; who controls the operations of the system

5. DESIGN / SOLUTION/METHODOLOGY :

5.1. DATABASE DESIGN :

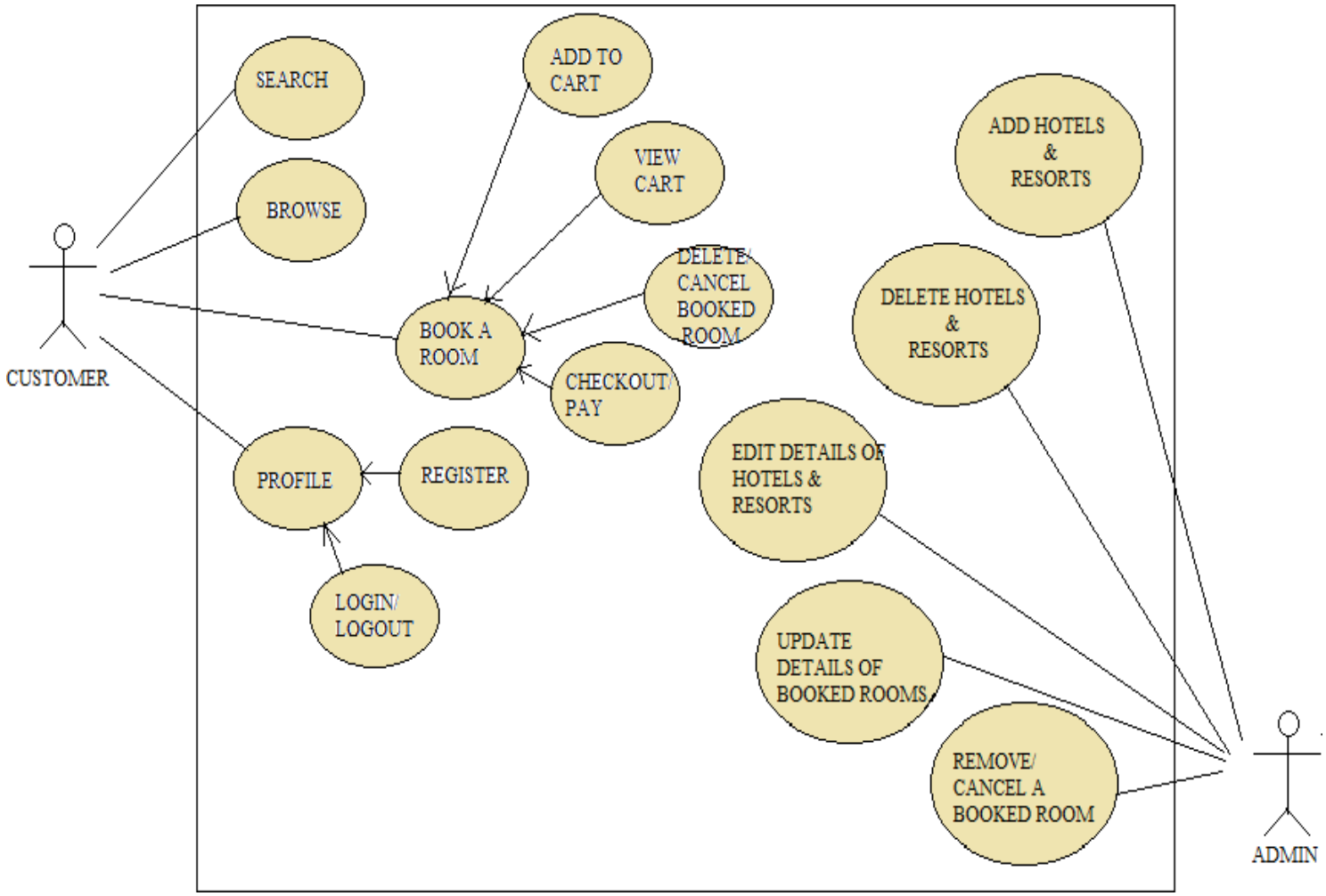
TABLE STRUCTURES / DATA DICTIONARY :



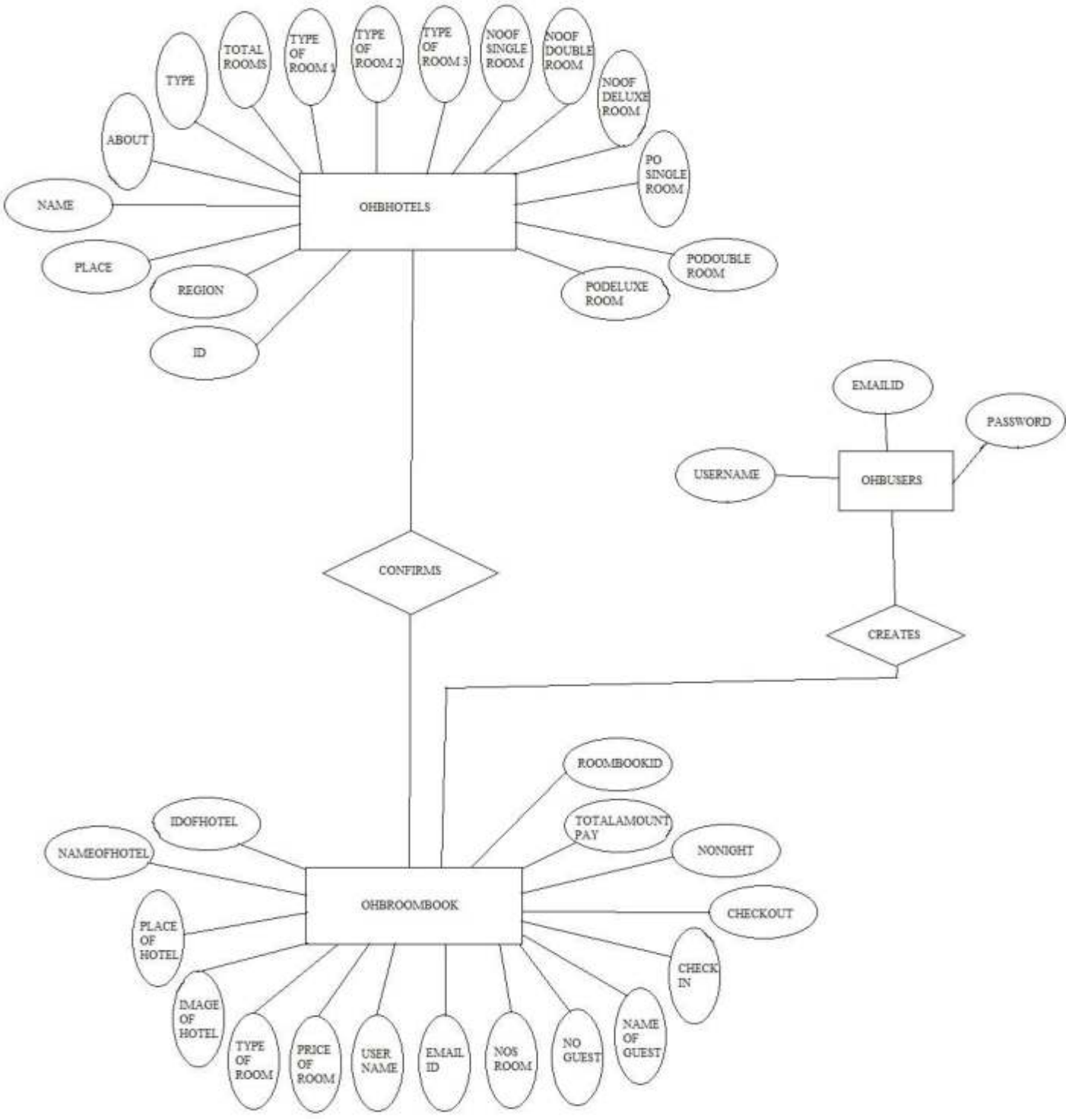

Oracle SQL*Plus
File Edit Search Options Help
SQL> desc ohbroombook;
Name Null? Type
IDOFHOTEL NUMBER(4)
NAMEOFHOTEL VARCHAR2(400)
PLACEOFHOTEL VARCHAR2(50)
IMAGEOFHOTEL VARCHAR2(200)
TYPEOFROOM VARCHAR2(10)
PRICEOFROOM NUMBER(7,2)
USERNAME VARCHAR2(20)
EMAILID VARCHAR2(20)
NOSROOM NUMBER(3)
NOGUEST NUMBER(3)
NAMEOFGUEST VARCHAR2(500)
CHECKIN VARCHAR2(20)
CHECKOUT VARCHAR2(20)
NONIGHT NUMBER(3)
TOTALAMOUNTPAY NUMBER(10,2)
ROOMBOOKID NUMBER(5)
SQL> desc ohbblog;
Name Null? Type
NAME VARCHAR2(100)
EMAILID VARCHAR2(100)
MESSAGE VARCHAR2(500)
SQL> desc ohbusers;
Name Null? Type
USERNAME VARCHAR2(20)
EMAILID VARCHAR2(20)
PASSWORD VARCHAR2(20)
SQL> desc ohbadmin;
Name Null? Type
USERID VARCHAR2(20)
PASSWORD VARCHAR2(20)
SQL>

5.2. ARCHITECTURAL DESIGN :

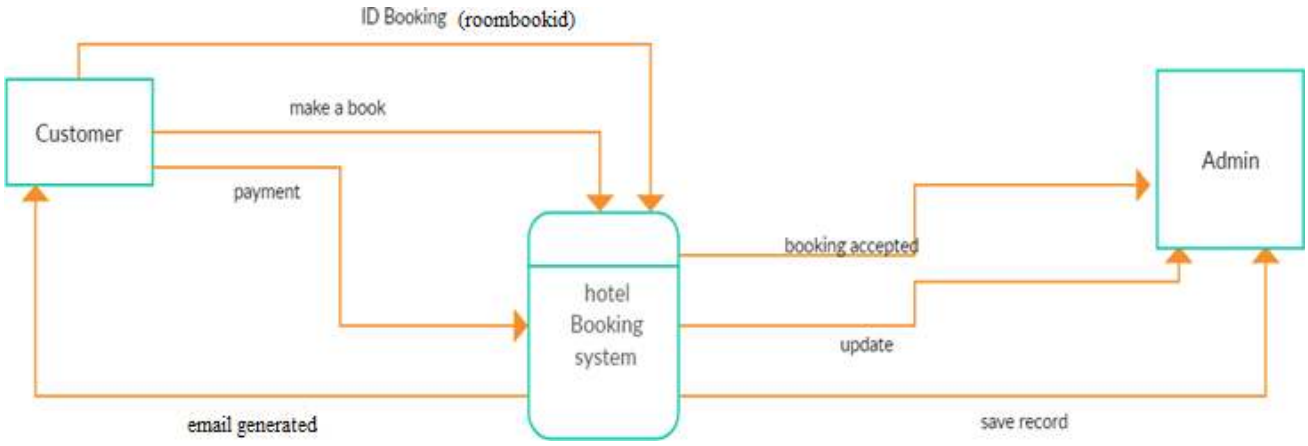
(USE CASE DIAGRAM) :



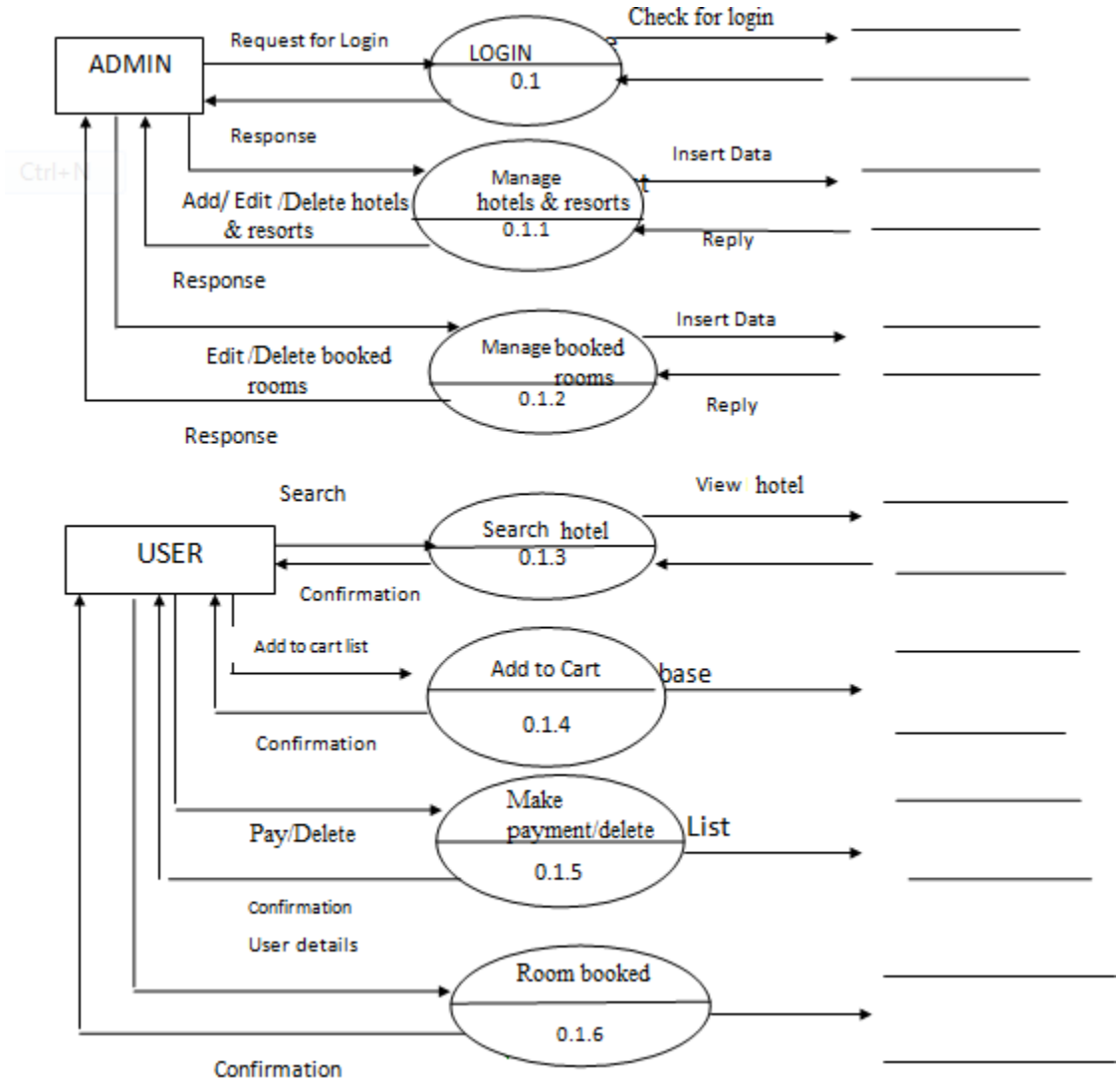
5. ENTITY RELATIONSHIP DIAGRAM (ERD) :



6. DATA FLOW DIAGRAM (DFD) :



LEVEL 0: DFD



LEVEL 1 : DFD

9. DEVELOPMENT:

After designing the structures of all the modules and database, the project is developed using, MVC(Model View Controller) Architecture. Implementation of this architecture separates the three parts from each other- Data, View and Business Logic and hence they can be developed separately. The View designed using JSP and the logic is embedded in the controller which is nothing but a servlet. In this project, we will be using pure JAVA Classes to hold the data. This pure JAVA classes will be having only attributes, variables representing corresponding data fields with no methods. The overall development is done and executed with the help of web server like Tomcat.

JAVA - OVERVIEW

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere.**

Java is −

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

- **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

- **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.

- **Distributed** − Java is designed for the distributed environment of the internet.

- **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

Evolution of Java

Java was initially launched as Java 1.0 but soon after its initial release, Java 1.1 was launched. Java 1.1 redefined event handling, new library elements were added.

In **Java 1.2** Swing and Collection framework was added
and `suspend()`, `resume()` and `stop()` methods were deprecated from **Thread** class.

No major changes were made into **Java 1.3** but the next release that was **Java 1.4** contained several important changes. Keyword `assert`, chained exceptions and channel based I/O System was introduced.

Java 1.5 was called **J2SE 5**, it added following major new features :

- Generics
- Annotations
- Autoboxing and autounboxing
- Enumerations
- For-each Loop
- Varargs
- Static Import
- Formatted I/O
- Concurrency utilities

Next major release was **Java SE 7** which included many new changes, like :

- Now **String** can be used to control Switch statement.
- Multi Catch Exception
- *try-with-resource* statement
- Binary Integer Literals
- *Underscore* in numeric literals, etc.

And the latest addition to the lot is, **Java SE 8**, it was released on March 18, 2014. Some of the major new features introduced in JAVA 8 are,

- Lambda Expressions
- New Collection Package java.util.stream to provide Stream API.
- Enhanced Security
- Nashorn Javascript Engine included
- Parallel Array Sorting
- The JDBC-ODBC Bridge has been removed etc.

Application of Java

Java is widely used in every corner of world and of human life. Java is not only used in software's but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java :

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems

Java Server Pages (JSP) Overview

What is Java Server Pages?

Java Server Pages (JSP) is a technology for developing WebPages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A Java Server Pages component is a type of Java servlet that is designed to fulfil the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create WebPages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

Why Use JSP?

Java Server Pages often serve the same purpose as programs implemented using the **Common Gateway Interface (CGI)**. But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- Java Server Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including **JDBC, JNDI, EJB, JAXP,** etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

Advantages of JSP

Following table lists out the other advantages of using JSP over other technologies −

vs. Active Server Pages (ASP)

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

vs. Pure Servlets

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

vs. Server-Side Includes (SSI)

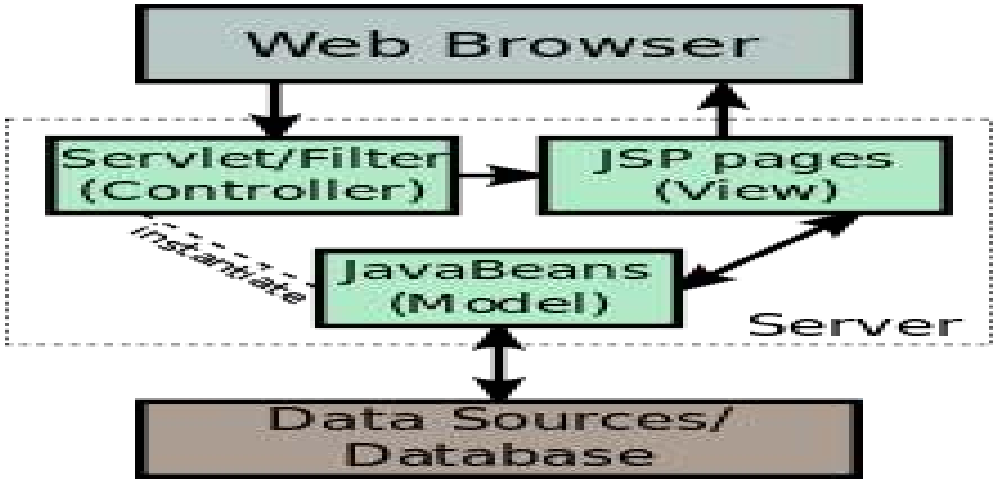
SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

vs. JavaScript

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

vs. Static HTML

Regular HTML, of course, cannot contain dynamic information.



As shown in above architecture, web browser directly accesses the JSP page of web container. The JSP pages interact with the web container's JavaBeans which represent the application model. When client sends request from a JSP page, the response is sent back to the client depending on the requests which are invoked by the client request. If response requires accessing the database, the JSP page uses JavaBeans to get data from the database.

Life Cycle of JSP

JSP pages follow these phases:

- Translate the JSP to servlet code.
- Compile the servlet to byte code.
- Load the servlet class.
- Create the servlet instance.
- Call the jspInit method.
- Call the _jspService method.
- Call the jspDestroy method.

JSP Page Translation: In the first step, the web container translates the JSP file into a Java source file that contains a servlet class definition. The web container validates the correctness of JSP pages and tag files.

JSP Page Compilation: In the second step, the web container compiles the servlet source code into a Java class file.

JSP Page Class Loading: In the third step, the servlet class byte code is loaded into the web container's JVM software using class loader.

JSP Page Servlet Instance: In the fourth step, the web container creates an instance of the servlet class.

JSP Page Initialization: In the fifth step, the web container initializes the servlet by calling the jspInit() method. This method is called immediately after the instance gets created. It is called only once during JSP life cycle.

To perform initialization we need to override jspInit() method:

```
public void jspInit ()
{ code }
```

JSP Page Service: The initialized servlet can now service the requests. The web container calls the _jspService () method to process the user request. This method is called for every request during its life cycle.

This method takes HttpServletRequest and HttpServletResponse parameters.

JSP Page Destroyed: When the web container removes the JSP servlet instance from service, it calls jspDestroy () method to allow JSP page to clean up the resources. This is end of the JSP life cycle.

This method can be written as follows:

```
public void jspDestroy()
{ code }
```

JSP Syntax

This topic defines about basic syntax in Java Server Page such as Scriptlet, Declarative, Expression and Comments.

Syntax Name	Code
Scriptlet Tag	<%statement or java code%>
Declarative tag	<%! Statement %>
Expression tag	<% = Statement %>
Comment tag	<%-----comments------ %>

The Scriptlet Tag: it is used to include java Technology Code in the JSP. The syntax is as follows:

```
<% java code %>
```

Listing 1. Example using scriptlet tag

```
<% int i=0; %>
```

The Declarative Tag: It is used to include members in the JSP servlet class either attributes or methods. The syntax is as follows:

```
<!  Statement %>
```

Listing 2. Example using declarative tag

```
<! Int counter=0; %>
```

The Expression Tag: It holds java language expression that is evaluated during HTTP request and is included in HTTP response stream. The syntax is as follows:

```
<%= expression %>
```

Listing 3. Example using expression tag

```
The current time and date is : <%= new java.util.Date () %>
```

The Comment Tag: It is used to hide part of code in JSP page or to ignore some statements in the code. The syntax is as follows:

```
<% - -comment -- %>
```

Listing 4. Example using comment tag

```
<%-- This is a JSP comment -- %>
```

Servlets - Overview

What are Servlets?

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

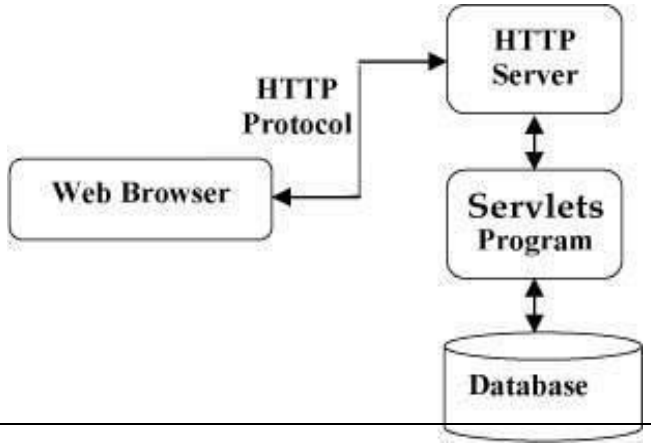
Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

- Performance is significantly better.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.
- Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

Servlets Architecture:

Following diagram shows the position of Servelts in a Web Application.



Servlets Tasks:

Servlets perform the following major tasks:

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

Servlets Packages:

Java Servlets are Java classes run by a web server that has an interpreter that supports the Java Servlet specification.

Servlets can be created using the **javax.servlet** and **javax.servlet.http** packages, which are a standard part of the Java's enterprise edition, an expanded version of the Java class library that supports large-scale development projects.

These classes implement the Java Servlet and JSP specifications. At the time of writing this tutorial, the versions are Java Servlet 2.5 and JSP 2.1.

Java servlets have been created and compiled just like any other Java class. After you install the servlet packages and add them to your computer's classpath, you can compile servlets with the JDK's Java compiler or any other current compiler.

JDBC Overview

The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.

JDBC helps you to write Java applications that manage these three programming activities:

1. Connect to a data source, like a database
2. Send queries and update statements to the database
3. Retrieve and process the results received from the database in answer to your query

The following simple code fragment gives a simple example of these three steps:

```
public void connectToAndQueryDatabase(String username, String password) {

   Connection con = DriverManager.getConnection(
                    "jdbc:myDriver:myDatabase",
                    username,
                    password);

   Statement stmt = con.createStatement();
   ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

   while (rs.next()) {
      int x = rs.getInt("a");
      String s = rs.getString("b");
      float f = rs.getFloat("c");
   }
}
```

This short code fragment instantiates a `DriverManager` object to connect to a database driver and log into the database, instantiates a `Statement` object that carries your SQL language query to the database; instantiates a `ResultSet` object that retrieves the results of your query, and executes a simple `while` loop, which retrieves and displays those results. It's that simple.

JDBC Product Components

JDBC includes four components:

1. **The JDBC API** — The JDBC™ API provides programmatic access to relational data from the Java™ programming language. Using the JDBC API, applications can execute SQL statements, retrieve results, and propagate changes back to an underlying data source. The JDBC API can also interact with multiple data sources in a distributed, heterogeneous environment.

   The JDBC API is part of the Java platform, which includes the *Java™ Standard Edition* (Java™ SE) and the *Java™ Enterprise Edition* (Java™ EE). The JDBC 4.0 API is divided into two packages: `java.sql` and `javax.sql`. Both packages are included in the Java SE and Java EE platforms.

2. **JDBC Driver Manager** — The JDBC `DriverManager` class defines objects which can connect Java applications to a JDBC driver. `DriverManager` has traditionally been the backbone of the JDBC architecture. It is quite small and simple.

   The Standard Extension packages `javax.naming` and `javax.sql` let you use a `DataSource` object registered with a *Java Naming and Directory Interface™* (JNDI) naming service to establish a connection with a data source. You can use either connecting mechanism, but using a `DataSource` object is recommended whenever possible.

3. **JDBC Test Suite** — The JDBC driver test suite helps you to determine that JDBC drivers will run your program. These tests are not comprehensive or exhaustive, but they do exercise many of the important features in the JDBC API.

4. **JDBC-ODBC Bridge** — The Java Software bridge provides JDBC access via ODBC drivers. Note that you need to load ODBC binary code onto each client machine that uses this driver. As a result, the ODBC driver is most appropriate on a corporate network where client installations are not a major problem, or for application server code written in Java in a three-tier architecture.

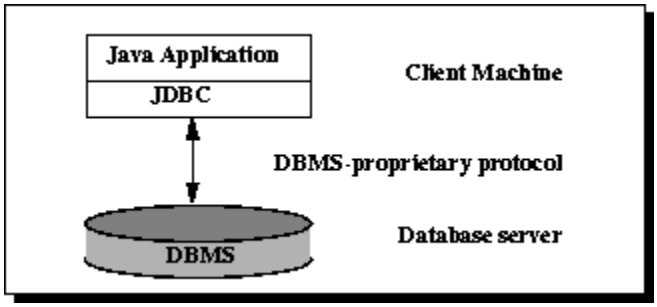
This Trail uses the first two of these these four JDBC components to connect to a database and then build a java program that uses SQL commands to communicate with a test Relational Database. The last two components are used in specialized environments to test web applications, or to communicate with ODBC-aware DBMSs.

JDBC Architecture

Two-tier and Three-tier Processing Models

The JDBC API supports both two-tier and three-tier processing models for database access.

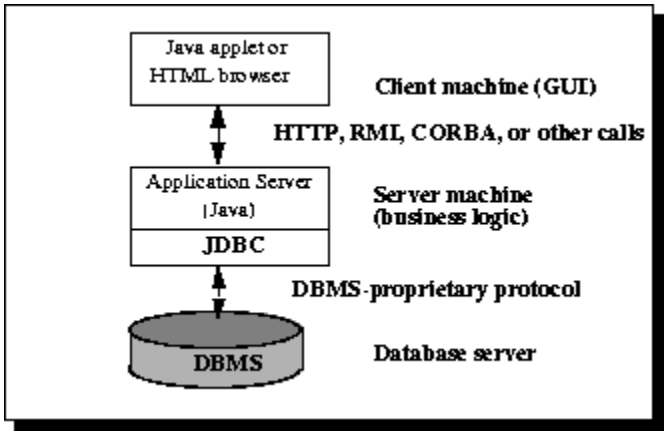
Figure 1: Two-tier Architecture for Data Access.



In the two-tier model, a Java applet or application talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed. A user's commands are delivered to the database or other data source, and the results of those statements are sent back to the user. The data source may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the data source as the server. The network can be an intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then sends the commands to the data source. The data source processes the commands and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that it simplifies the deployment of applications. Finally, in many cases, the three-tier architecture can provide performance advantages.

Figure 2: Three-tier Architecture for Data Access.



Until recently, the middle tier has often been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code and technologies such as Enterprise JavaBeans™, the Java platform is fast becoming the standard platform for middle-tier development. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features.

A RELATIONAL DATABASE OVERVIEW

A database is a means of storing information in such a way that information can be retrieved from it. In simplest terms, a relational database is one that presents information in tables with rows and columns. A table is referred to as a relation in the sense that it is a collection of objects of the same type (rows). Data in a table can be related according to common keys or concepts, and the ability to retrieve related data from a table is the basis for the term relational database. A Database Management System (DBMS) handles the way data is stored, maintained, and retrieved. In the case of a relational database, a Relational Database Management System (RDBMS) performs these tasks. DBMS as used in this book is a general term that includes RDBMS.

Integrity Rules

Relational tables follow certain integrity rules to ensure that the data they contain stay accurate and are always accessible. First, the rows in a relational table should all be distinct. If there are duplicate rows, there can be problems resolving which of two possible selections is the correct one. For most DBMSs, the user can specify that duplicate rows are not allowed, and if that is done, the DBMS will prevent the addition of any rows that duplicate an existing row.

A second integrity rule of the traditional relational model is that column values must not be repeating groups or arrays. A third aspect of data integrity involves the concept of a null value. A database takes care of situations where data may not be available by using a null value to indicate that a value is missing. It does not equate to a blank or zero. A blank is considered equal to another blank, a zero is equal to another zero, but two null values are not considered equal.

When each row in a table is different, it is possible to use one or more columns to identify a particular row. This unique column or group of columns is called a primary key. Any column that is part of a primary key cannot be null; if it were, the primary key containing it would no longer be a complete identifier. This rule is referred to as entity integrity.

The `Employees` table illustrates some of these relational database concepts. It has five columns and six rows, with each row representing a different employee.

`Employees` Table

Employee_Number	First_name	Last_Name	Date_of_Birth	Car_Number
10001	Axel	Washington	28-Aug-43	5
10083	Arvid	Sharma	24-Nov-54	Null
10120	Jonas	Ginsberg	01-Jan-69	Null
10005	Florence	Wojokowski	04-Jul-71	12
10099	Sean	Washington	21-Sep-66	Null
10035	Elizabeth	Yamaguchi	24-Dec-59	Null

The primary key for this table would generally be the employee number because each one is guaranteed to be different. (A number is also more efficient than a string for making comparisons.) It would also be possible to use `First_Name` and `Last_Name` because the combination of the two also identifies just one row in our sample database. Using the last name alone would not work because there are two employees with the last name of "Washington." In this particular case the first names are all different, so one could conceivably use that column as a primary key, but it is best to avoid using a column where duplicates could occur. If Elizabeth Yamaguchi gets a job at this company and the primary key is `First_Name`, the RDBMS will not allow her name to be added (if it has been specified that no duplicates are permitted). Because there is already an Elizabeth in the table, adding a second one would make the primary key useless as a way of identifying just one row. Note that although using `First_Name` and `Last_Name` is a unique composite key for this example, it might not be unique in a larger database. Note also that the `Employee` table assumes that there can be only one car per employee.

SELECT Statements

SQL is a language designed to be used with relational databases. There is a set of basic SQL commands that is considered standard and is used by all RDBMSs. For example, all RDBMSs use the `SELECT` statement.

A `SELECT` statement, also called a query, is used to get information from a table. It specifies one or more column headings, one or more tables from which to select, and some criteria for selection. The RDBMS returns rows of the column entries that satisfy the stated requirements. A `SELECT` statement such as the following will fetch the first and last names of employees who have company cars:

```
SELECT First_Name, Last_Name
FROM Employees
WHERE Car_Number IS NOT NULL
```

The result set (the set of rows that satisfy the requirement of not having null in the `Car_Number` column) follows. The first name and last name are printed for each row that satisfies the requirement because the `SELECT` statement (the first line) specifies the columns `First_Name` and `Last_Name`. The `FROM` clause (the second line) gives the table from which the columns will be selected.

FIRST_NAME	LAST_NAME
Axel	Washington
Florence	Wojokowski

The following code produces a result set that includes the whole table because it asks for all of the columns in the table Employees with no restrictions (no `WHERE` clause). Note that `SELECT *` means "SELECT all columns."

```
SELECT *
```

FROM Employees

WHERE Clauses

The `WHERE` clause in a `SELECT` statement provides the criteria for selecting values. For example, in the following code fragment, values will be selected only if they occur in a row in which the column Last_Name begins with the string 'Washington'.

```
SELECT First_Name, Last_Name
FROM Employees
WHERE Last_Name LIKE 'Washington%'
```

The keyword `LIKE` is used to compare strings, and it offers the feature that patterns containing wildcards can be used. For example, in the code fragment above, there is a percent sign (`%`) at the end of 'Washington', which signifies that any value containing the string 'Washington' plus zero or more additional characters will satisfy this selection criterion. So 'Washington' or 'Washingtonian' would be matches, but 'Washing' would not be. The other wildcard used in `LIKE` clauses is an underbar (_), which stands for any one character. For example,

```
WHERE Last_Name LIKE 'Ba_man'
```

would match 'Batman', 'Barman', 'Badman', 'Balman', 'Bagman', 'Bamman', and so on.

The code fragment below has a `WHERE` clause that uses the equal sign (=) to compare numbers. It selects the first and last name of the employee who is assigned car 12.

```
SELECT First_Name, Last_Name
FROM Employees
WHERE Car_Number = 12
```

The next code fragment selects the first and last names of employees whose employee number is greater than 10005:

```
SELECT First_Name, Last_Name
FROM Employees
```

```
WHERE Employee_Number > 10005
```

`WHERE` clauses can get rather elaborate, with multiple conditions and, in some DBMSs, nested conditions. This overview will not cover complicated `WHERE` clauses, but the following code fragment has a `WHERE` clause with two conditions; this query selects the first and last names of employees whose employee number is less than 10100 and who do not have a company car.

```
SELECT First_Name, Last_Name
FROM Employees
WHERE Employee_Number < 10100 and Car_Number IS NULL
```

A special type of `WHERE` clause involves a join, which is explained in the next section.

JOINS

A distinguishing feature of relational databases is that it is possible to get data from more than one table in what is called a join. Suppose that after retrieving the names of employees who have company cars, one wanted to find out who has which car, including the make, model, and year of car. This information is stored in another table, `Cars`:

`Cars` Table

Car_Number	Make	Model	Year
5	Honda	Civic DX	1996
12	Toyota	Corolla	1999

There must be one column that appears in both tables in order to relate them to each other. This column, which must be the primary key in one table, is called the foreign key in the other table. In this case, the column that appears in two tables is `Car_Number`, which is the primary key for the table `Cars` and the foreign key in the table Employees. If the 1996 Honda Civic were wrecked and deleted from the `Cars` table, then `Car_Number` 5 would also have to be

removed from the Employees table in order to maintain what is called referential integrity. Otherwise, the foreign key column (`Car_Number`) in the `Employees` table would contain an entry that did not refer to anything in `Cars`. A foreign key must either be null or equal to an existing primary key value of the table to which it refers. This is different from a primary key, which may not be null. There are several null values in the `Car_Number` column in the table `Employees` because it is possible for an employee not to have a company car.

The following code asks for the first and last names of employees who have company cars and for the make, model, and year of those cars. Note that the `FROM` clause lists both Employees and Cars because the requested data is contained in both tables. Using the table name and a dot (.) before the column name indicates which table contains the column.

```
SELECT Employees.First_Name, Employees.Last_Name,
   Cars.Make, Cars.Model, Cars.Year
FROM Employees, Cars
WHERE Employees.Car_Number = Cars.Car_Number
```

This returns a result set that will look similar to the following:

FIRST_NAME	LAST_NAME	MAKE	MODEL	YEAR
Axel	Washington	Honda	Civic DX	1996
Florence	Wojokowski	Toyota	Corolla	1999

Common SQL Commands

SQL commands are divided into categories, the two main ones being Data Manipulation Language (DML) commands and Data Definition Language (DDL) commands. DML commands deal with data, either retrieving it or modifying it to keep it up-to-date. DDL commands create or change tables and other database objects such as views and indexes.

A list of the more common DML commands follows:

- `SELECT` – used to query and display data from a database. The `SELECT` statement specifies which columns to include in the result set. The vast majority of the SQL commands used in applications are `SELECT` statements.
- `INSERT` – adds new rows to a table. `INSERT` is used to populate a newly created table or to add a new row (or rows) to an already-existing table.
- `DELETE` – removes a specified row or set of rows from a table
- `UPDATE` – changes an existing value in a column or group of columns in a table

The more common DDL commands follow:

- `CREATE TABLE` – creates a table with the column names the user provides. The user also needs to specify a type for the data in each column. Data types vary from one RDBMS to another, so a user might need to use metadata to establish the data types used by a particular database. `CREATE TABLE` is normally used less often than the data manipulation commands because a table is created only once, whereas adding or deleting rows or changing individual values generally occurs more frequently.
- `DROP TABLE` – deletes all rows and removes the table definition from the database. A JDBC API implementation is required to support the `DROP TABLE` command as specified by SQL92, Transitional Level. However, support for the `CASCADE` and `RESTRICT` options of `DROP TABLE` is optional. In addition, the behavior of `DROP TABLE` is implementation-defined when there are views or integrity constraints defined that reference the table being dropped.
- `ALTER TABLE` – adds or removes a column from a table. It also adds or drops table constraints and alters column attributes

Result Sets and Cursors

The rows that satisfy the conditions of a query are called the result set. The number of rows returned in a result set can be zero, one, or many. A user can access the data in a result set one row at a time, and a cursor provides the means to do that. A cursor can be thought of as a pointer into a file that contains the rows of the result set, and that pointer has the ability to keep track of which row is currently being accessed. A cursor allows a user to process each row of a

result set from top to bottom and consequently may be used for iterative processing. Most DBMSs create a cursor automatically when a result set is generated.

Earlier JDBC API versions added new capabilities for a result set's cursor, allowing it to move both forward and backward and also allowing it to move to a specified row or to a row whose position is relative to another row.

Transactions

When one user is accessing data in a database, another user may be accessing the same data at the same time. If, for instance, the first user is updating some columns in a table at the same time the second user is selecting columns from that same table, it is possible for the second user to get partly old data and partly updated data. For this reason, DBMSs use transactions to maintain data in a consistent state (data consistency) while allowing more than one user to access a database at the same time (data concurrency).

A transaction is a set of one or more SQL statements that make up a logical unit of work. A transaction ends with either a commit or a rollback, depending on whether there are any problems with data consistency or data concurrency. The commit statement makes permanent the changes resulting from the SQL statements in the transaction, and the rollback statement undoes all changes resulting from the SQL statements in the transaction.

A lock is a mechanism that prohibits two transactions from manipulating the same data at the same time. For example, a table lock prevents a table from being dropped if there is an uncommitted transaction on that table. In some DBMSs, a table lock also locks all of the rows in a table. A row lock prevents two transactions from modifying the same row, or it prevents one transaction from selecting a row while another transaction is still modifying it.

Stored Procedures

A stored procedure is a group of SQL statements that can be called by name. In other words, it is executable code, a mini-program, that performs a particular task that can be invoked the same way one can call a function or method. Traditionally, stored procedures have been

written in a DBMS-specific programming language. The latest generation of database products allows stored procedures to be written using the Java programming language and the JDBC API. Stored procedures written in the Java programming language are byte code portable between DBMSs. Once a stored procedure is written, it can be used and reused because a DBMS that supports stored procedures will, as its name implies, store it in the database.

The following code is an example of how to create a very simple stored procedure using the Java programming language. Note that the stored procedure is just a static Java method that contains normal JDBC code. It accepts two input parameters and uses them to change an employee's car number.

Do not worry if you do not understand the example at this point. The code example below is presented only to illustrate what a stored procedure looks like. You will learn how to write the code in this example in the tutorials that follow.

```
import java.sql.*;
public class UpdateCar {
   public static void UpdateCarNum(int carNo, int empNo) throws SQLException {
      Connection con = null;
      PreparedStatement pstmt = null;
      try {
         con = DriverManager.getConnection("jdbc:default:connection");
         pstmt = con.prepareStatement(
                  "UPDATE EMPLOYEES " +
                  "SET CAR_NUMBER = ? " +
                  "WHERE EMPLOYEE_NUMBER = ?");
         pstmt.setInt(1, carNo);
         pstmt.setInt(2, empNo);
         pstmt.executeUpdate();
      }
      finally {
         if (pstmt != null) pstmt.close();
```

```
        }
    }
}
```

9.TESTING:

Before the implementation, function of the whole project needs to be tested for ensuring proper navigation and valid data entry into the forms and corresponding actions accordingly. This is only after a throughout testing process, when the project is ready to be implemented and to be used by the client in the real environment.

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Types of Testing

Different types of testing that may be used to test software during SDLC.

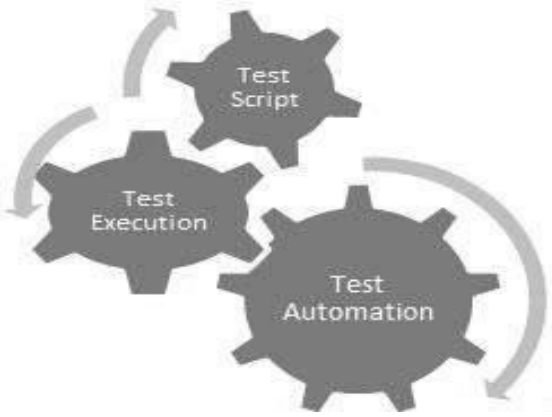
Manual Testing

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

Automation Testing

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.



Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

Software Testing – Methods

There are different methods that can be used for software testing. This chapter briefly describes the methods available.

Black-Box Testing :

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

The following table lists the advantages and disadvantages of black-box testing.

Advantages	Disadvantages
• Well suited and efficient for large code segments. • Code access is not required. • Clearly separates user's perspective from the developer's perspective through visibly defined roles. • Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems.	• Limited coverage, since only a selected number of test scenarios is actually performed. • Inefficient testing, due to the fact that the tester only has limited knowledge about an application. • Blind coverage, since the tester cannot target specific code segments or error-prone areas. • The test cases are difficult to design.

White-Box Testing :

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

The following table lists the advantages and disadvantages of white-box testing.

Advantages	Disadvantages
• As the tester has knowledge of the source code, it becomes very easy	• Due to the fact that a skilled tester is needed to perform white-box testing,

to find out which type of data can help in testing the application effectively. • It helps in optimizing the code. • Extra lines of code can be removed which can bring in hidden defects. • Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.	the costs are increased. • Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested. • It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools.

Grey-Box Testing :

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

Advantages	Disadvantages
• Offers combined benefits of black-box and white-box testing wherever possible. • Grey box testers don't rely on the source code; instead they rely on	• Since the access to source code is not available, the ability to go over the code and test coverage is limited. • The tests can be redundant if the

interface definition and functional specifications. • Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling. • The test is done from the point of view of the user and not the designer.	software designer has already run a test case. • Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

A Comparison of Testing Methods:

The following table lists the points that differentiate black-box testing, grey-box testing, and white-box testing.

Black-Box Testing	Grey-Box Testing	White-Box Testing
The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledge of the internal workings of the application.
Also known as closed-box testing, data-driven testing, or functional testing.	Also known as translucent testing, as the tester has limited knowledge of the insides of the application.	Also known as clear-box testing, structural testing, or code-based testing.
Performed by end-users and also by testers and developers.	Performed by end-users and also by testers and developers.	Normally done by testers and developers.

Testing is based on external expectations - Internal behavior of the application is unknown.	Testing is done on the basis of high-level database diagrams and data flow diagrams.	Internal workings are fully known and the tester can design test data accordingly.

IMPLEMENTATION AND MAINTENANCE:

This is the last stage in SDLC. The project is implemented for the client in real environment where it can be used. The project then runs in the real environment with all the real process executed by the user. The implementation includes the proper development of the project on to the server. After the real execution of the project under the real circumstances by the user, some loop holes may be uncovered. It is also possible that after some time the user may request some changes or report some problems in the working system. The maintenance of a project means updating the project for the required changes in the implemented system.

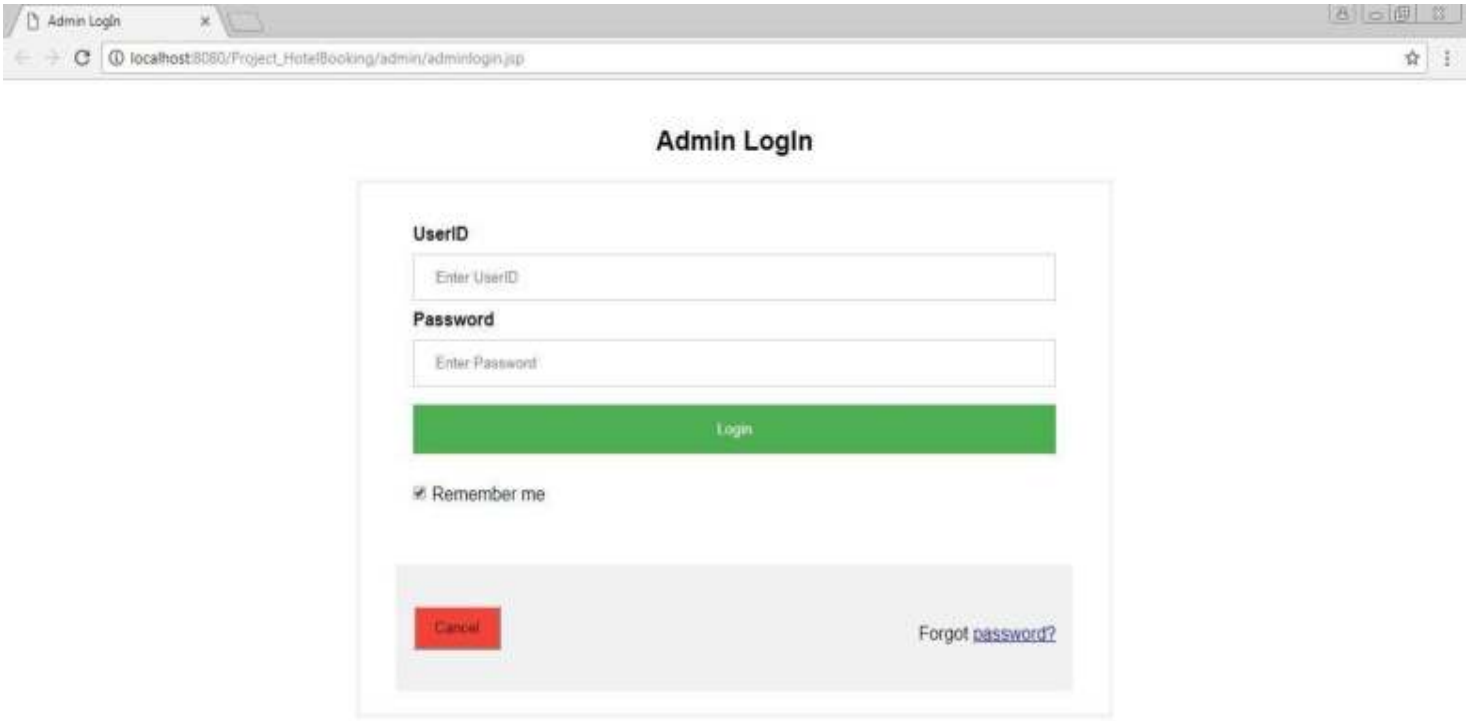
PHYSICAL DESIGN

ADMIN PART :

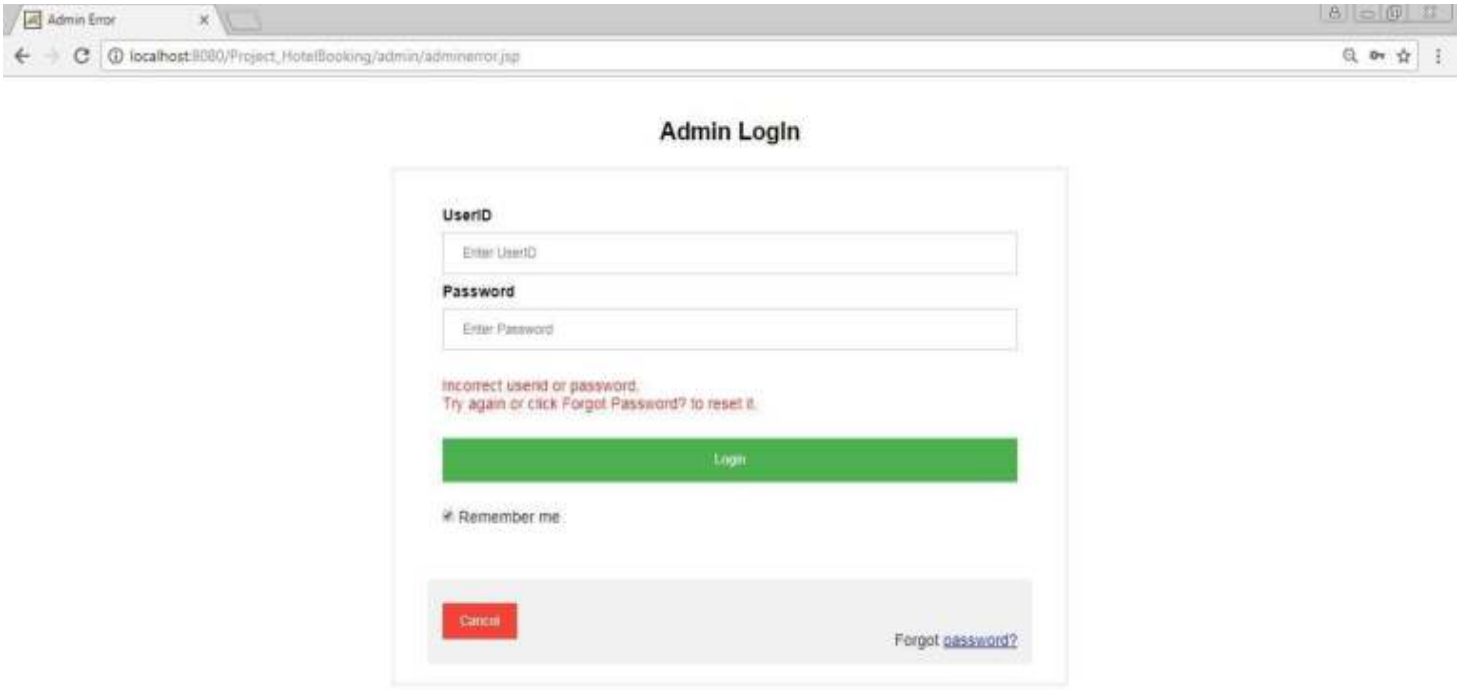
The first part that has been developed is the admin part since the administrator is going to enter, update and delete values from the database.

So the first is the **Admin Login part** :

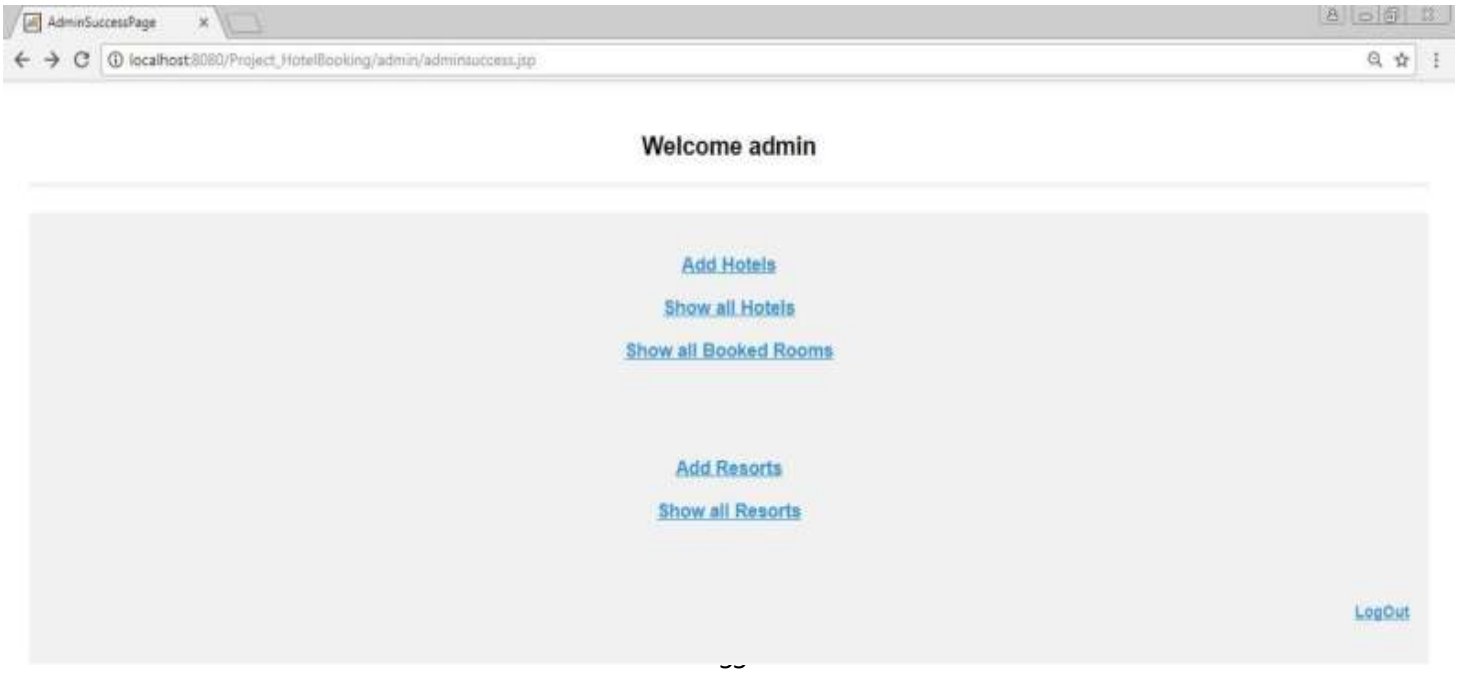
1) Below is the **Admin Login Page('adminlogin.jsp').** All the admin related pages has been kept in a separate folder called 'admin' under Web Content.



2) The UserID and Password to login into the administrator account is 'admin' and 'admin'. If the admin enters the wrong userid or password by mistake he/she is denied access to the admin account and the admin is redirected to **'adminerror.jsp' page** where the error message is displayed.



3) Now if the administrator enters the correct UserID and Password i.e., 'admin' and 'admin' he/she is granted access to the admin account and is redirected to the **'adminsuccess.jsp' page** where he/she can perform all admin related tasks such as adding new hotels, editing/updating the details of the exising hotels and deleting hotels. The admin can also view all the rooms that have been booked by different users in different hotels. The admin can edit/update those details and can also delete/cancel rooms that have been booked by users.

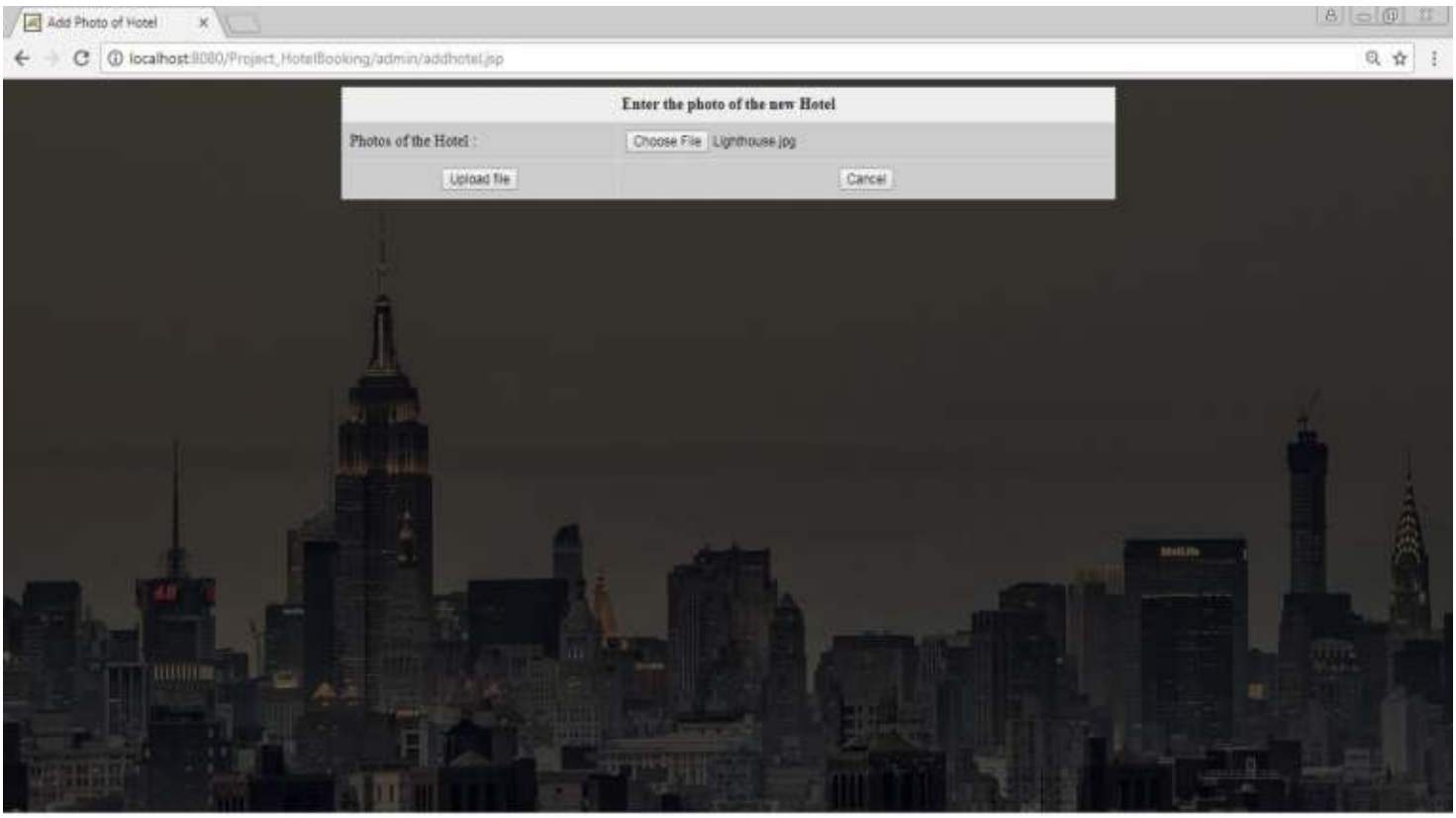


4) After finishing all the tasks if admin wants to 'LogOut' of his account he/she can click on the 'LogOut' link. Clicking on the **'LogOut' link** will get the admin logged out from his account and he will be directed to the login page**('adminlogin.jsp')** again.

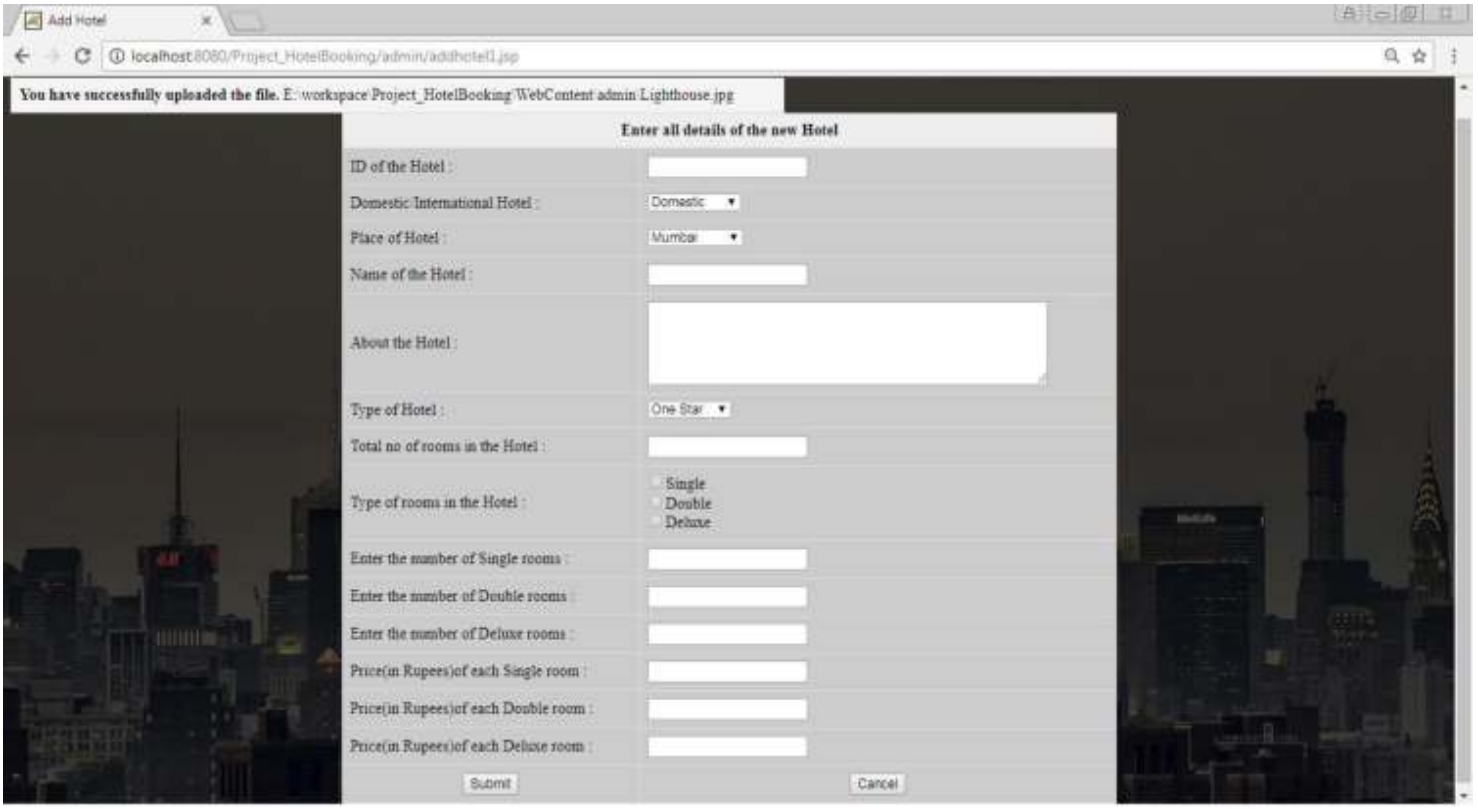
TASKS CARRIED OUT BY THE ADMIN

Adding/Updating/Deleting Hotels

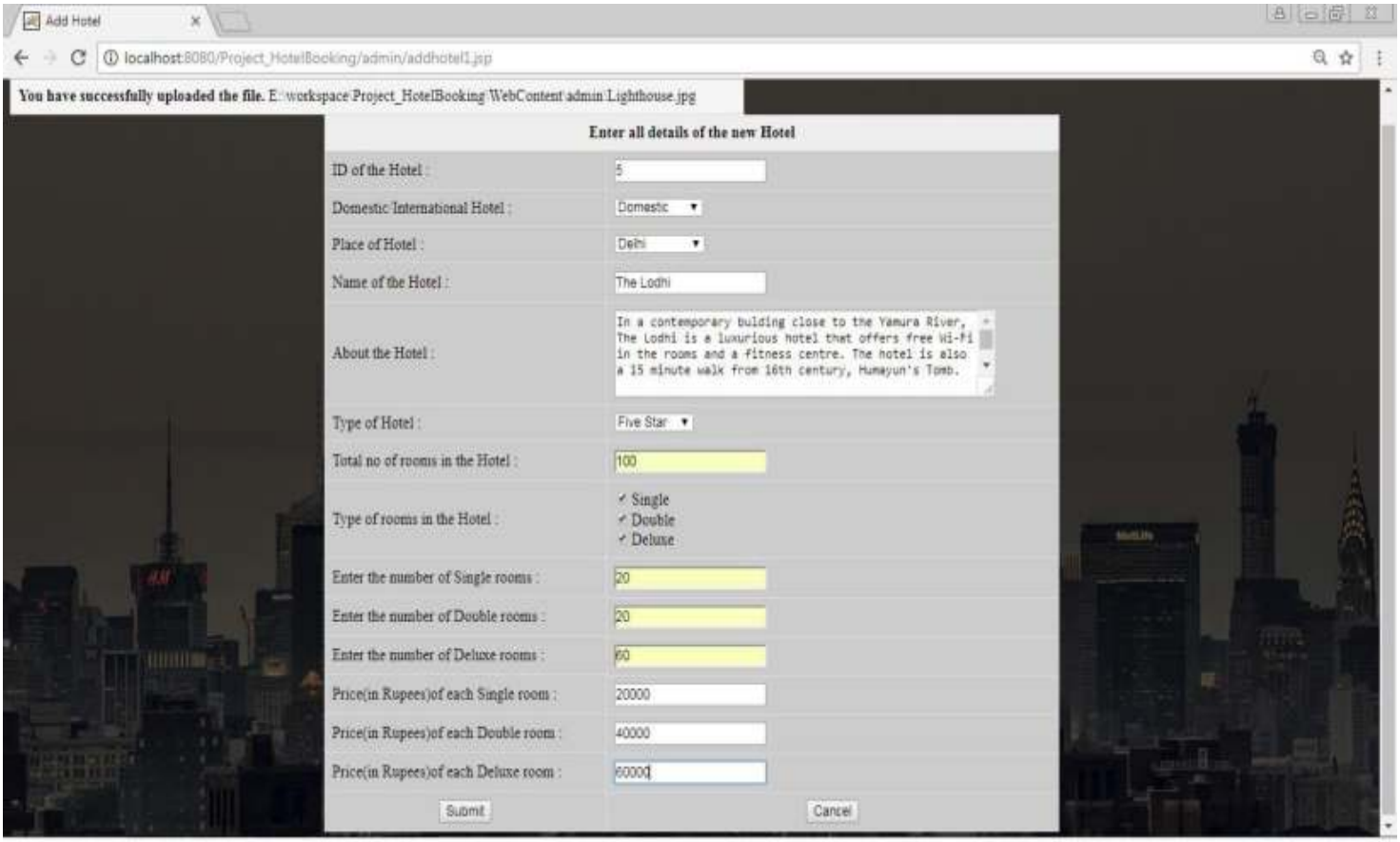
1) Now if the admin wants to add a new hotel in the database he has to click on the **'Add Hotels' link** present on the top of the **'adminsuccess.jsp'** page.
After clicking on the 'Add Hotels' link **'addhotel.jsp' page** opens where the admin has to first add the photo of the new hotel which he/she wants to add.



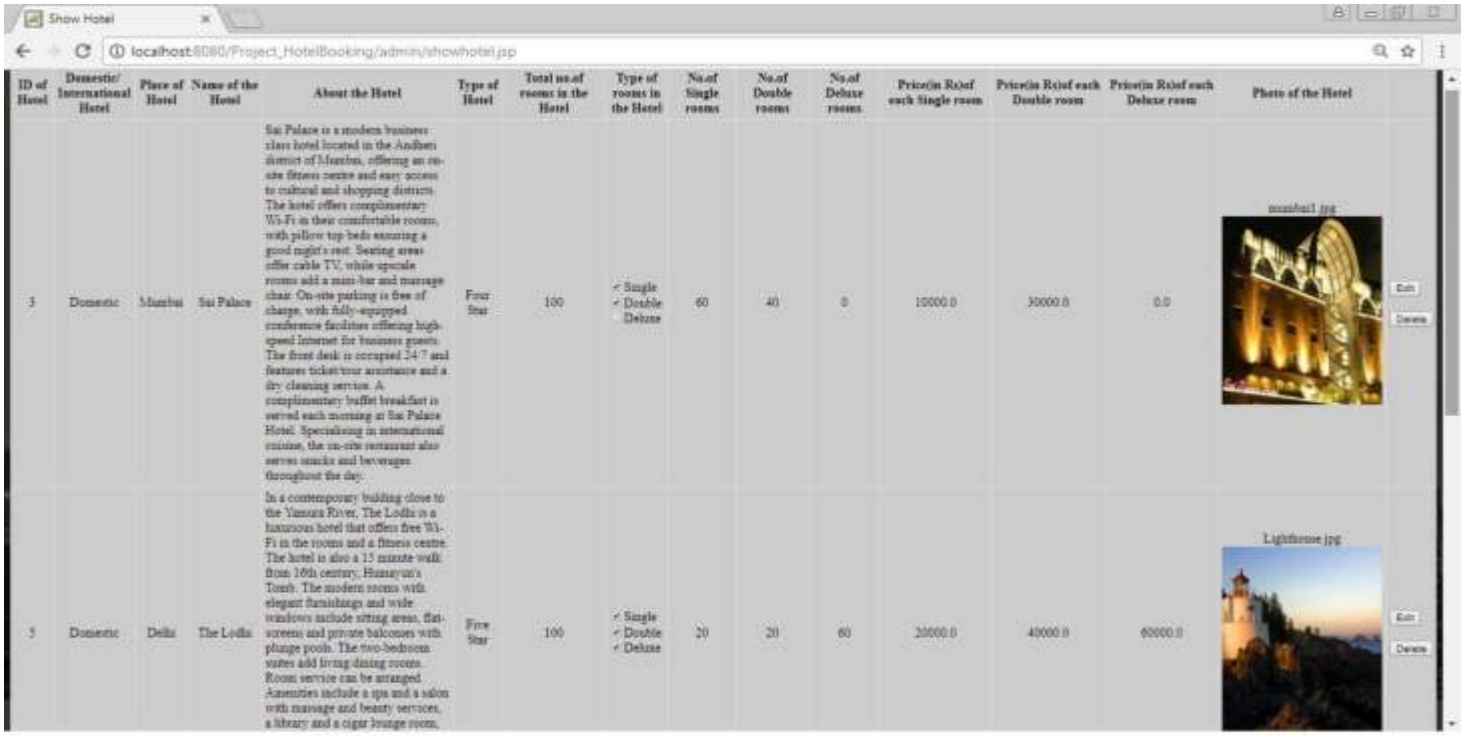
2) After adding the photo of the new hotel, the admin clicks on the 'Upload file' button after which the image path gets uploaded into the database and thereafter he/she is directed to a new page **'addhotel1.jsp'** where he/she has to enter all the details of the new hotel.



3) After entering all the hotel details the admin has to click on 'Submit' to submit all the details entered.



4) After clicking on 'Submit' all the details that the admin has entered gets inserted into the database 'ohbhotels' and all the data's so entered are displayed in a page **'showhotel.jsp'.**
The 'showhotel.jsp' page displays all the hotels present in the database.



There are two buttons at the end of each row – **'Edit'** and **'Delete'**. By clicking on the 'Edit' button the admin can update/edit the previous details of the hotels and by clicking on the 'Delete' button the admin can permanently remove the record of the hotel. There will be no existence of the hotel in the database if it is deleted by admin.

Editing/Deleting the details of rooms booked by users

The admin can also see all the rooms booked by users. He can see those by clicking on the **'Show all Booked Rooms' link** present in the **'adminsuccess.jsp' page**.
By clicking on that link he is directed to **'showbookedrooms.jsp' page** where he can see all the rooms that have been booked by users and also the user details. He can edit those details so entered by the users and can also delete or cancel a booking by clicking on the **'Edit'** and **'Delete'** buttons respectively present at the end of each row.



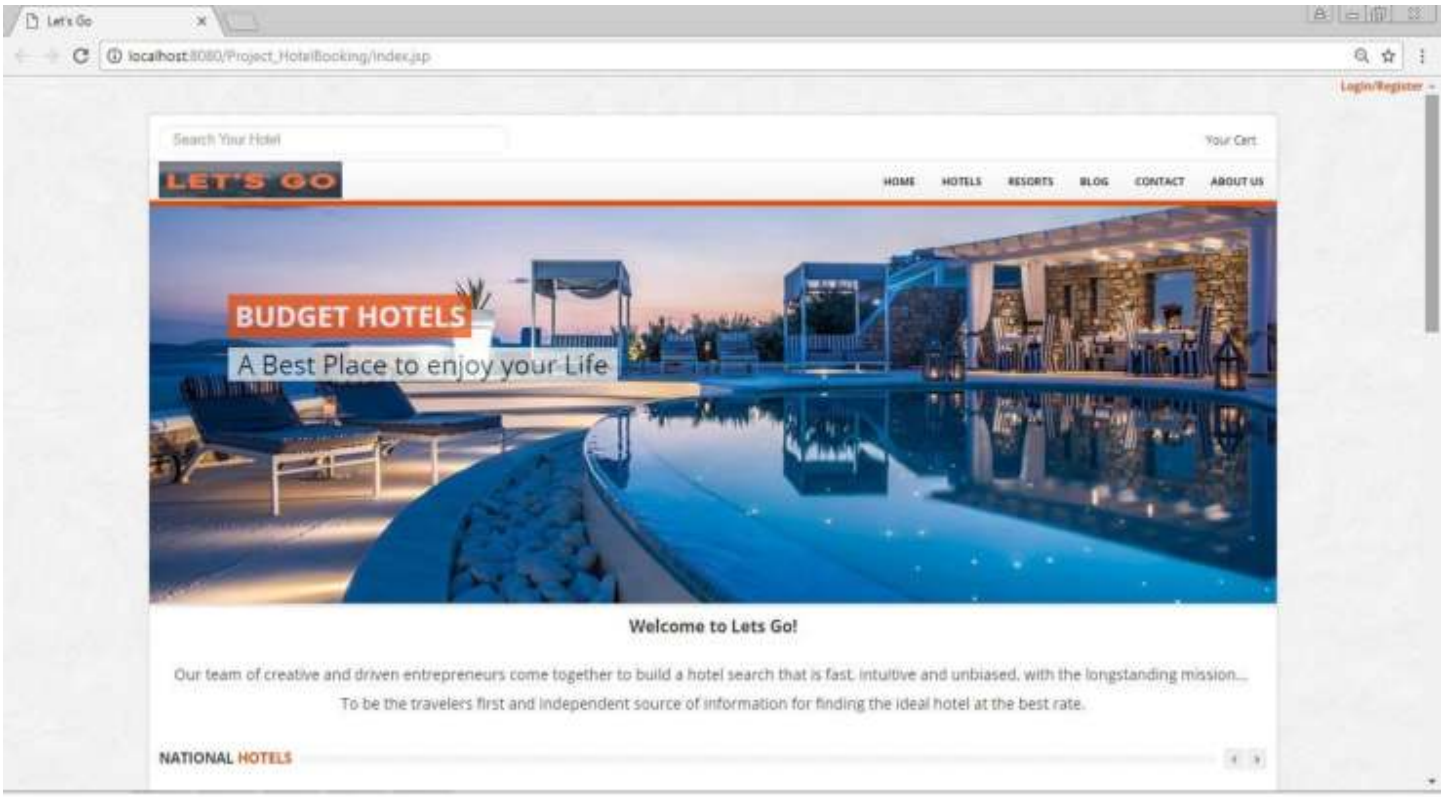
ID of Hotel	Name of Hotel	Place of Hotel	Type of room	Price of room/Night	Username	Email Id	No.of rooms booked	No.of Guests	Name of all Guests	Check-In Date	Check-Out Date	No of Nights	Total amount to be paid	Image of the Hotel	Room book Id	
1	ITC Maurya Sheraton	Delhi	Double	20000.0	Vishakha	kabasiv@gmail.com	1	1	A Kumar	2018-06-05	2018-06-27	22	440000.0	delhi1.jpg	161	Edit Delete
1	ITC Maurya Sheraton	Delhi	null	0.0	Vishakha	kabasiv@gmail.com	1	1	A Kumar	2018-05-29	2018-05-30	1	0.0	delhi1.jpg	202	Edit Delete
2	The Leela Palace	Delhi	Deluxe	40000.0	Vishakha	kabasiv@gmail.com	1	1	A Kumar	2018-05-29	2018-05-31	2	80000.0	delhi2.jpg	162	Edit Delete
	The													delhi2.jpg		Edit

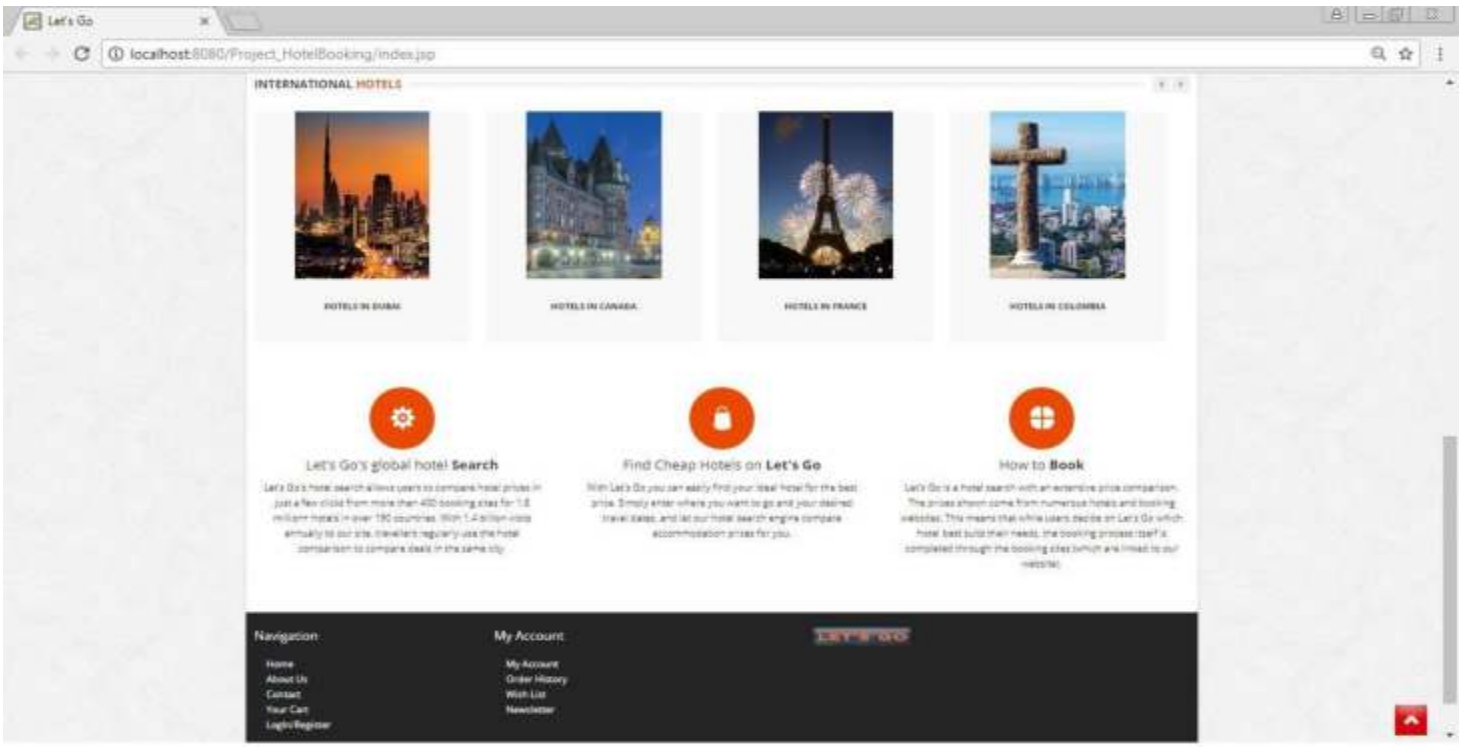
This is all about the Hotels part. The steps are same for the Resorts part so we didn't show it.

This is all about the 'Admin' part. Next is the User Interface part.

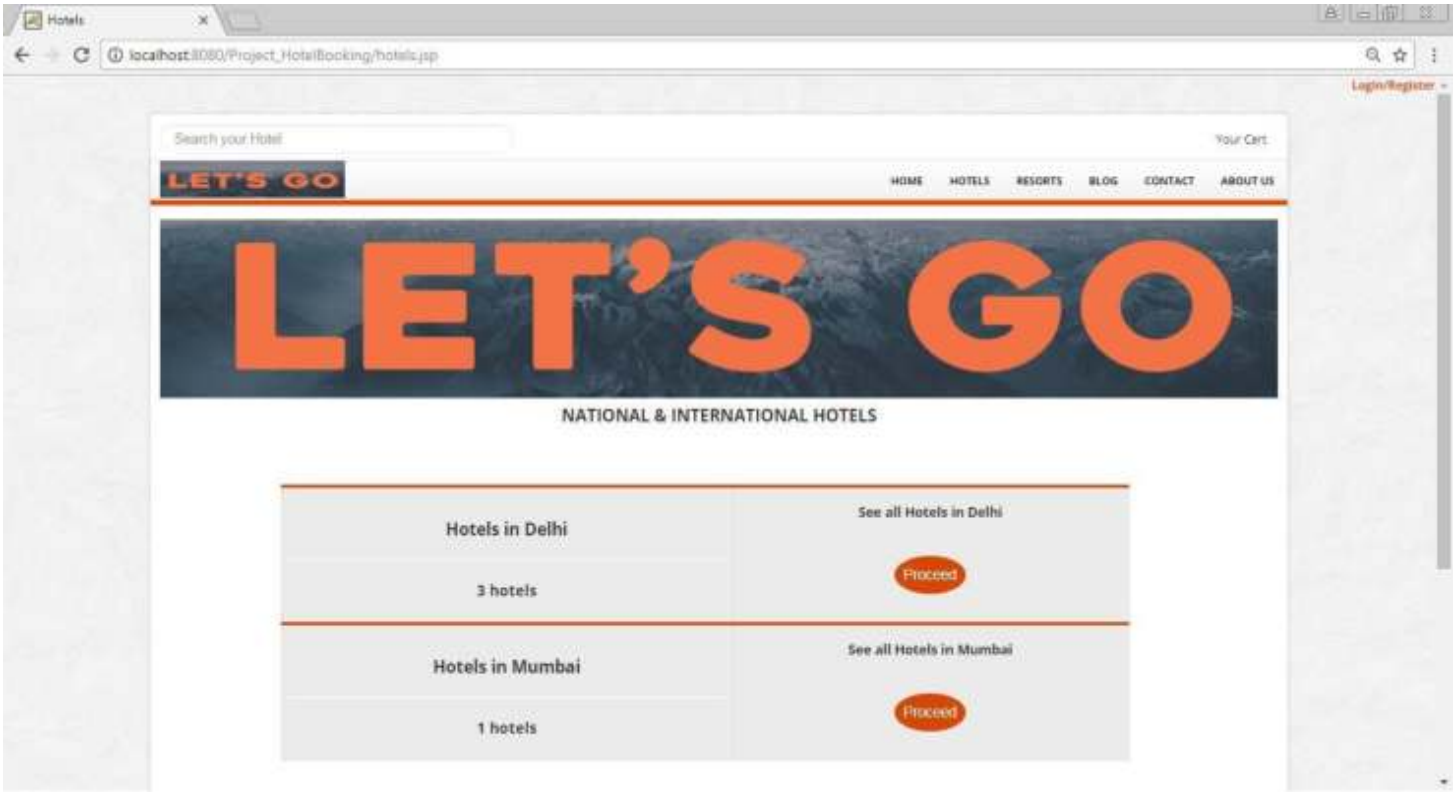
USER INTERFACE PART

1) The first comes the **Home page** or the **Index page (index.jsp).**

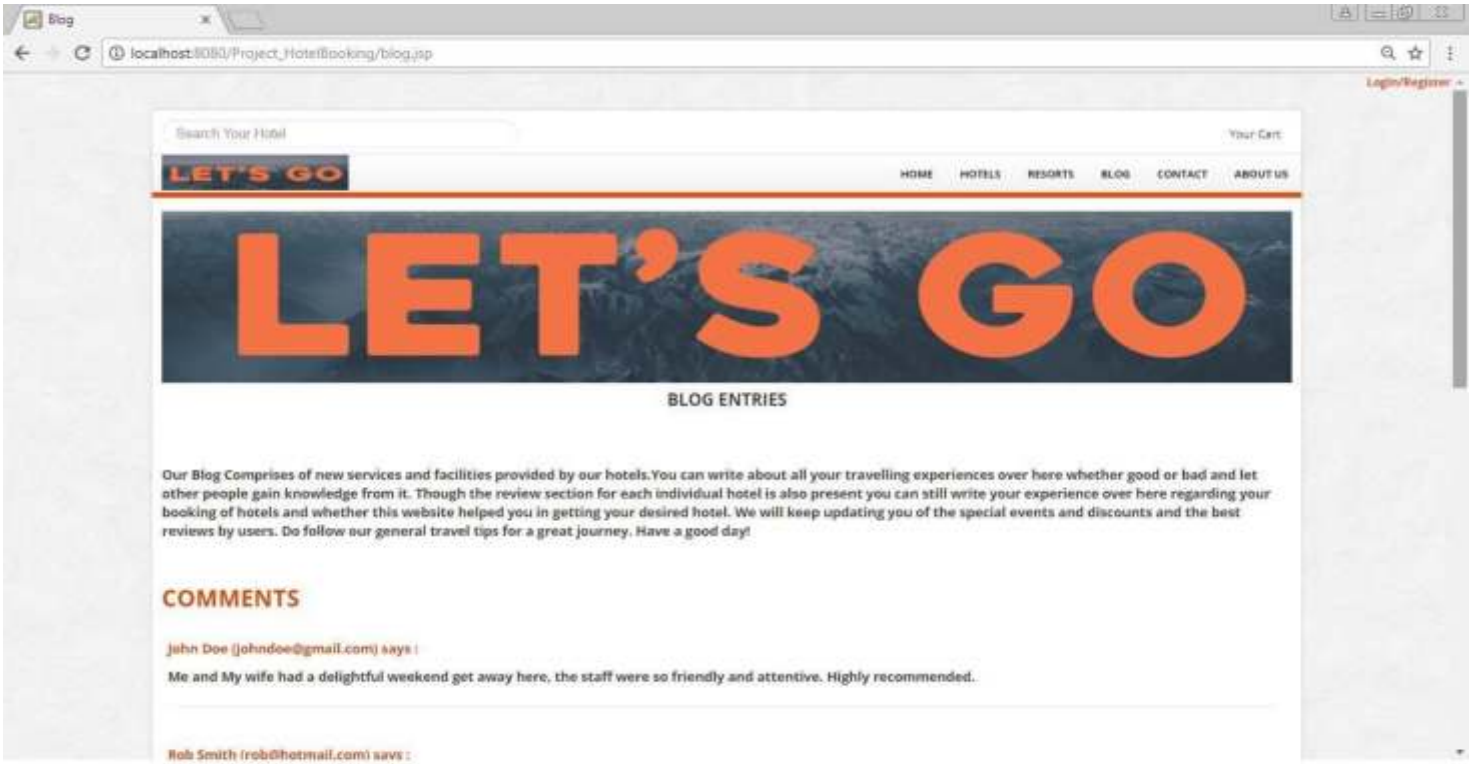


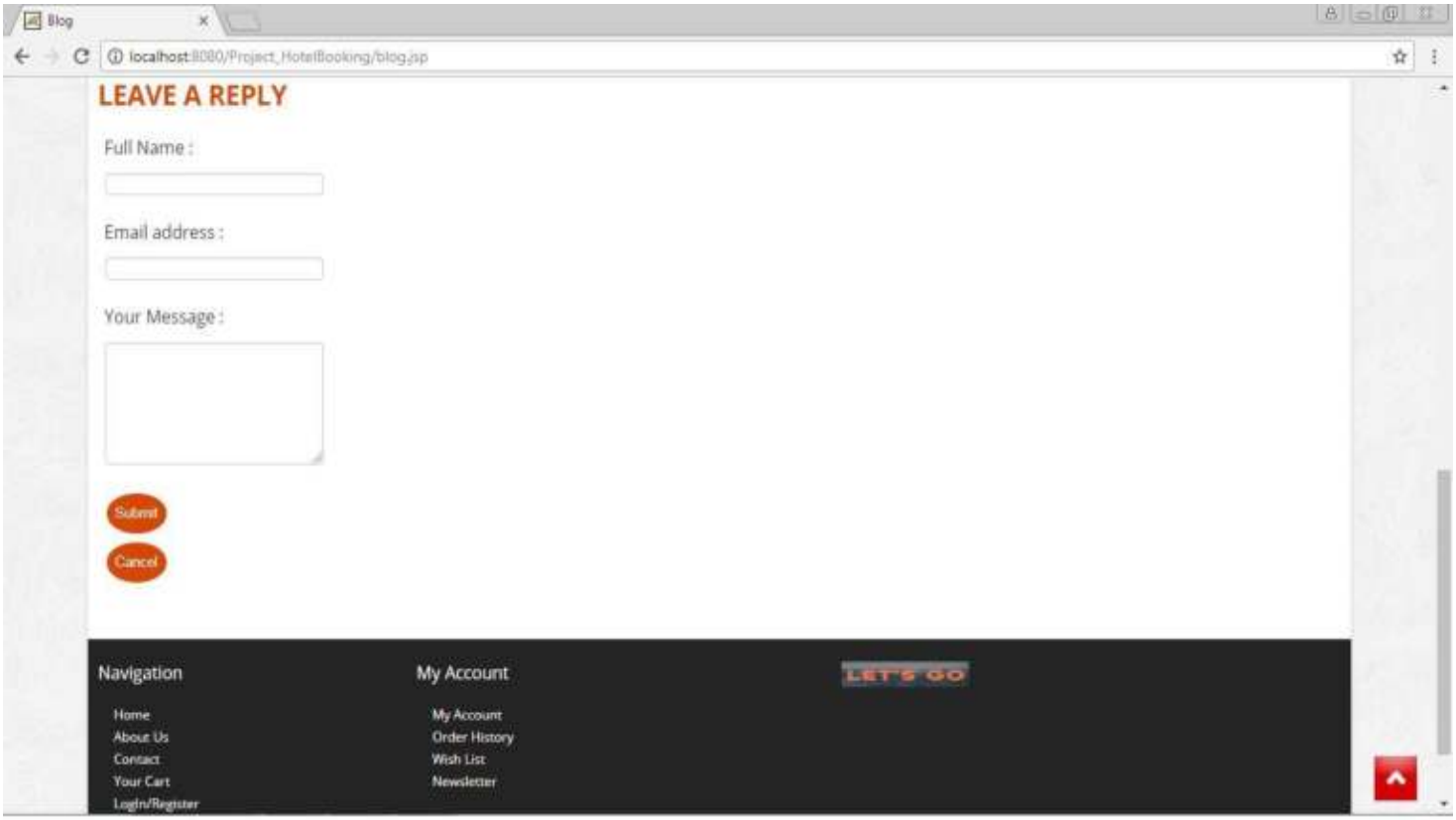


2) Clicking on the **'HOTELS'** menu in the **'index.jsp'** page takes the users to the **'hotels.jsp'** page where different places and the number of hotels in those places are displayed. The same goes for the 'RESORTS' menu. It takes the user to **'resorts.jsp'** page.

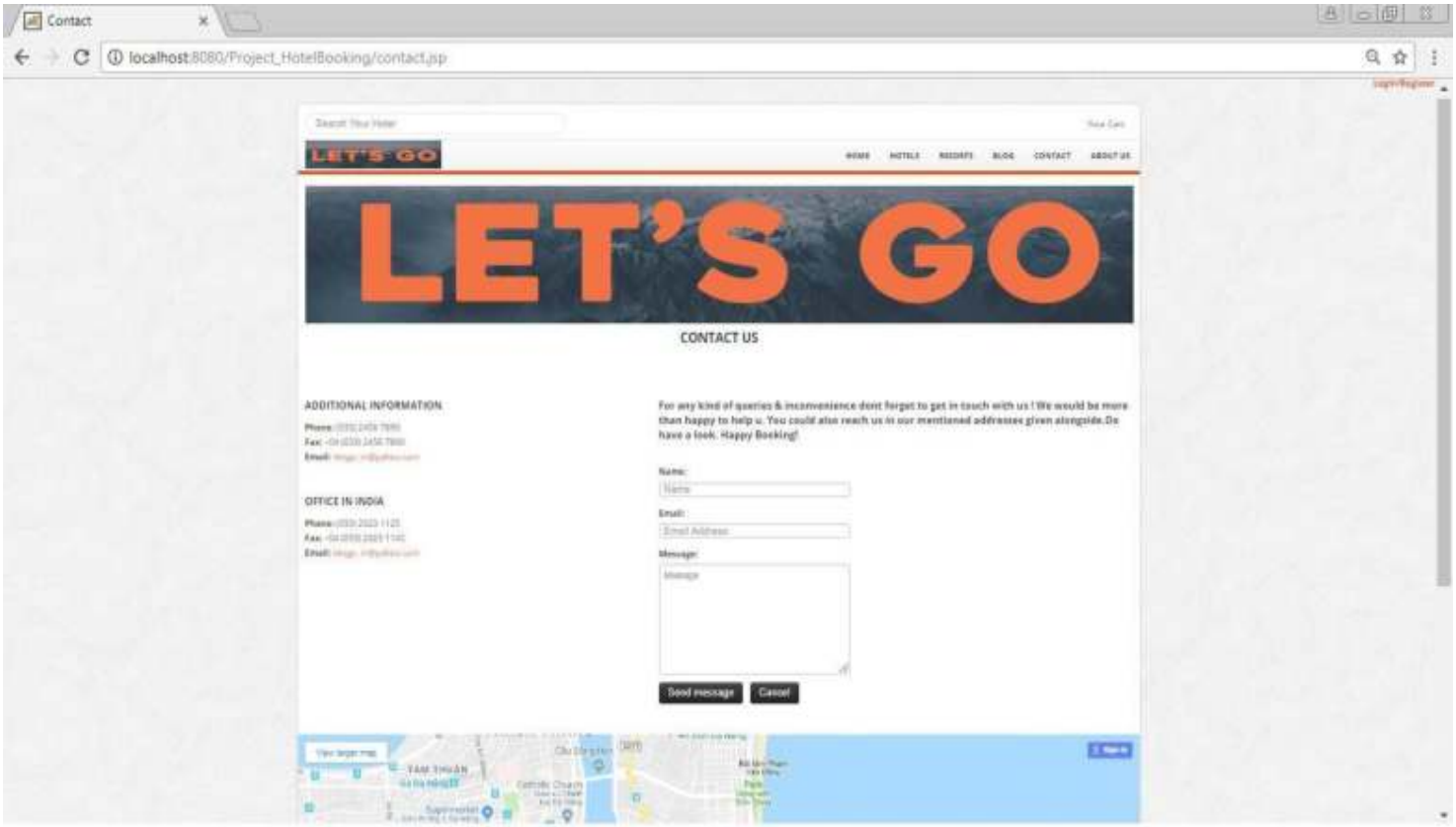


3) Clicking on the **'BLOG'** menu takes the user to **'blog.jsp'** page which shows all the feedbacks given by different users. The user can add a reply by filling in the details and then clicking on the 'Submit' button.


Blog
localhost:8080/Project_HotelBooking/blog.jsp
Login/Register
Search Your Hotel
Your Cart
LET'S GO
HOME
HOTELS
RESORTS
BLOG
CONTACT
ABOUT US
LET'S GO
BLOG ENTRIES
Our Blog Comprises of new services and facilities provided by our hotels.You can write about all your travelling experiences over here whether good or bad and let other people gain knowledge from it. Though the review section for each individual hotel is also present you can still write your experience over here regarding your booking of hotels and whether this website helped you in getting your desired hotel. We will keep updating you of the special events and discounts and the best reviews by users. Do follow our general travel tips for a great journey. Have a good day!
COMMENTS
John Doe (johndoe@gmail.com) says :
Me and My wife had a delightful weekend get away here, the staff were so friendly and attentive. Highly recommended.
Rob Smith (rob@hotmail.com) says :

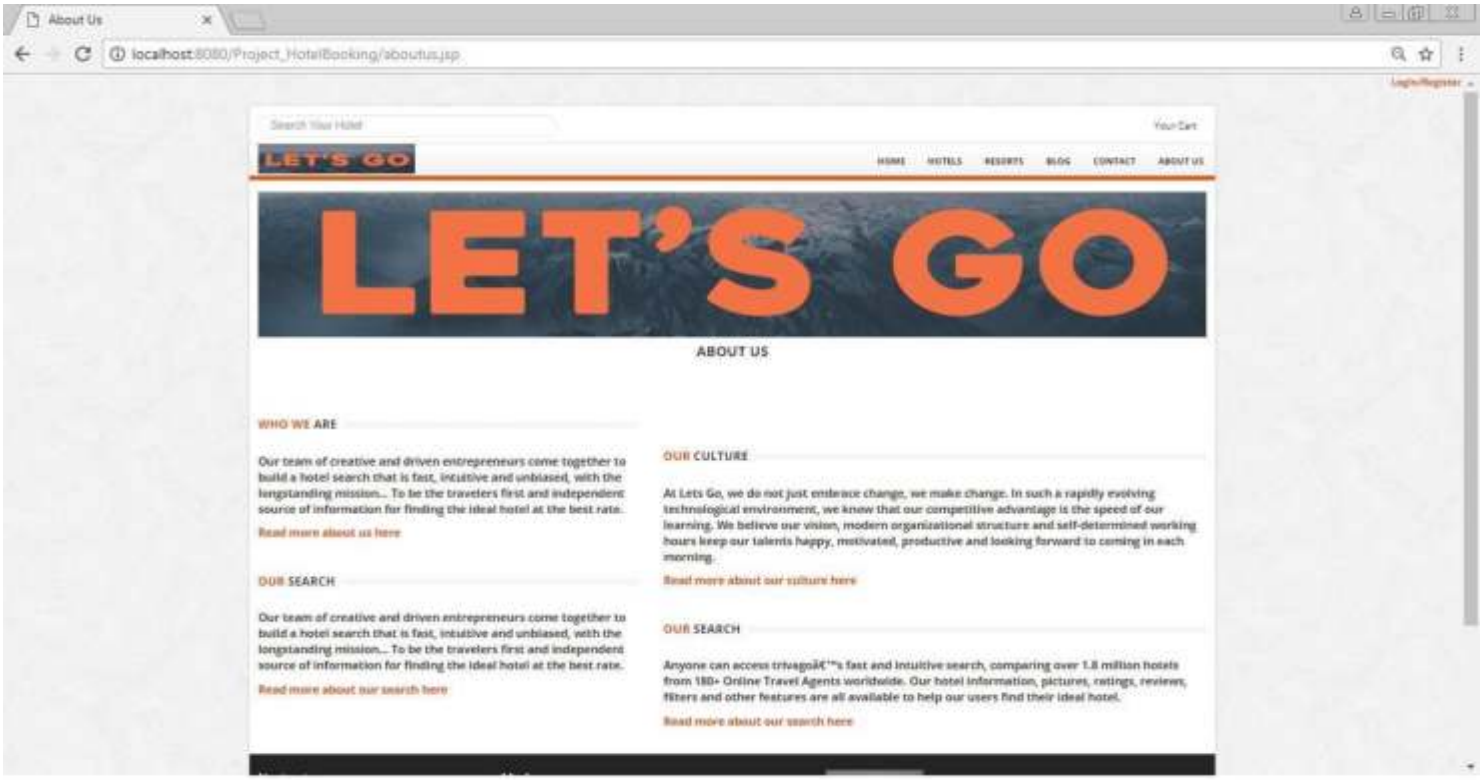

Blog
localhost:8080/Project_HotelBooking/blog.jsp
LEAVE A REPLY
Full Name :
Email address :
Your Message :
Submit
Cancel
Navigation
Home
About Us
Contact
Your Cart
Login/Register
My Account
My Account
Order History
Wish List
Newsletter
LET'S GO

4) Clicking on the **'CONTACT'** menu takes the user to **'contact.jsp' page** which shows the address and phone number of their office to contact them.

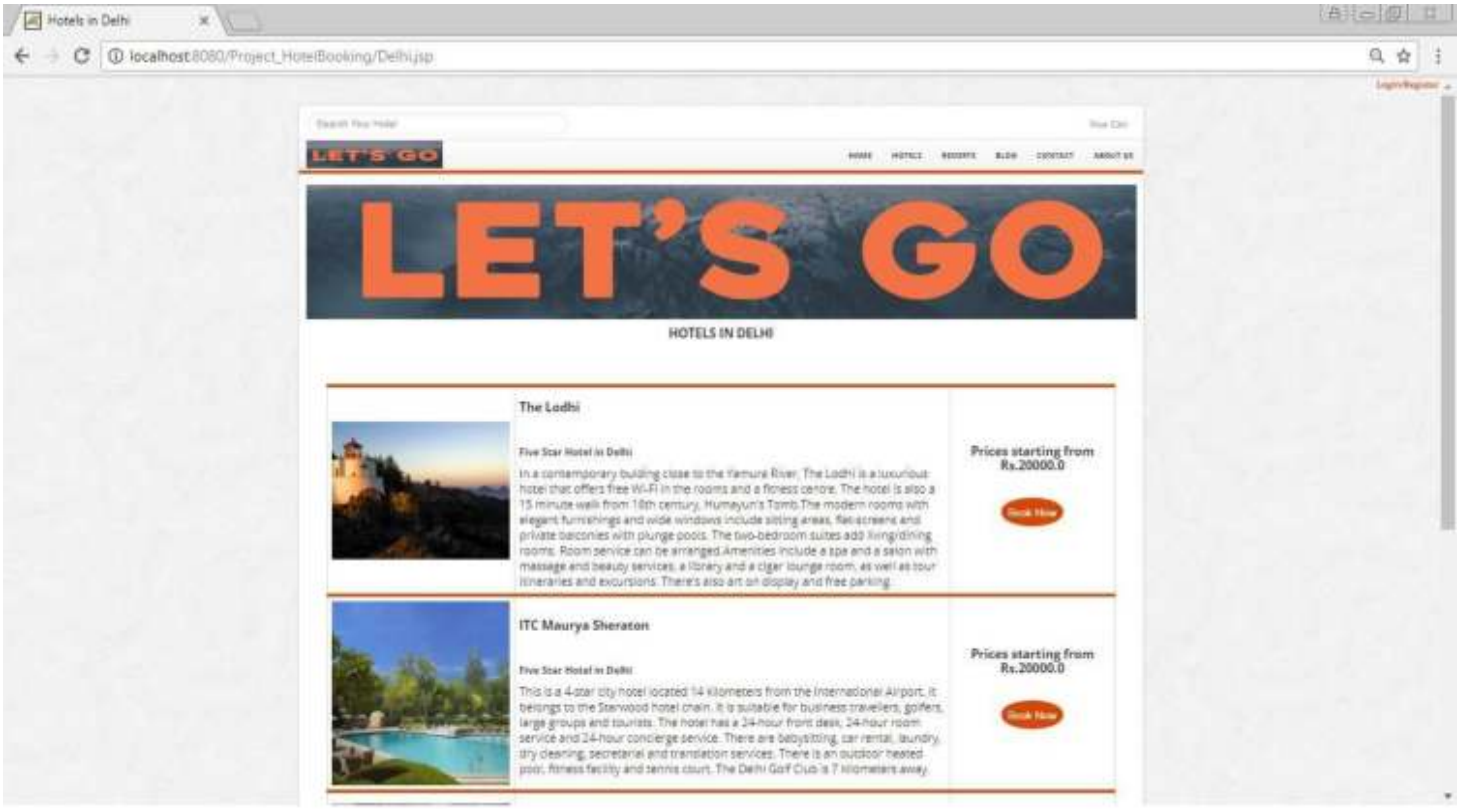


In case of any enquiry the user can contact the officials of Let's Go online by sending them a message.

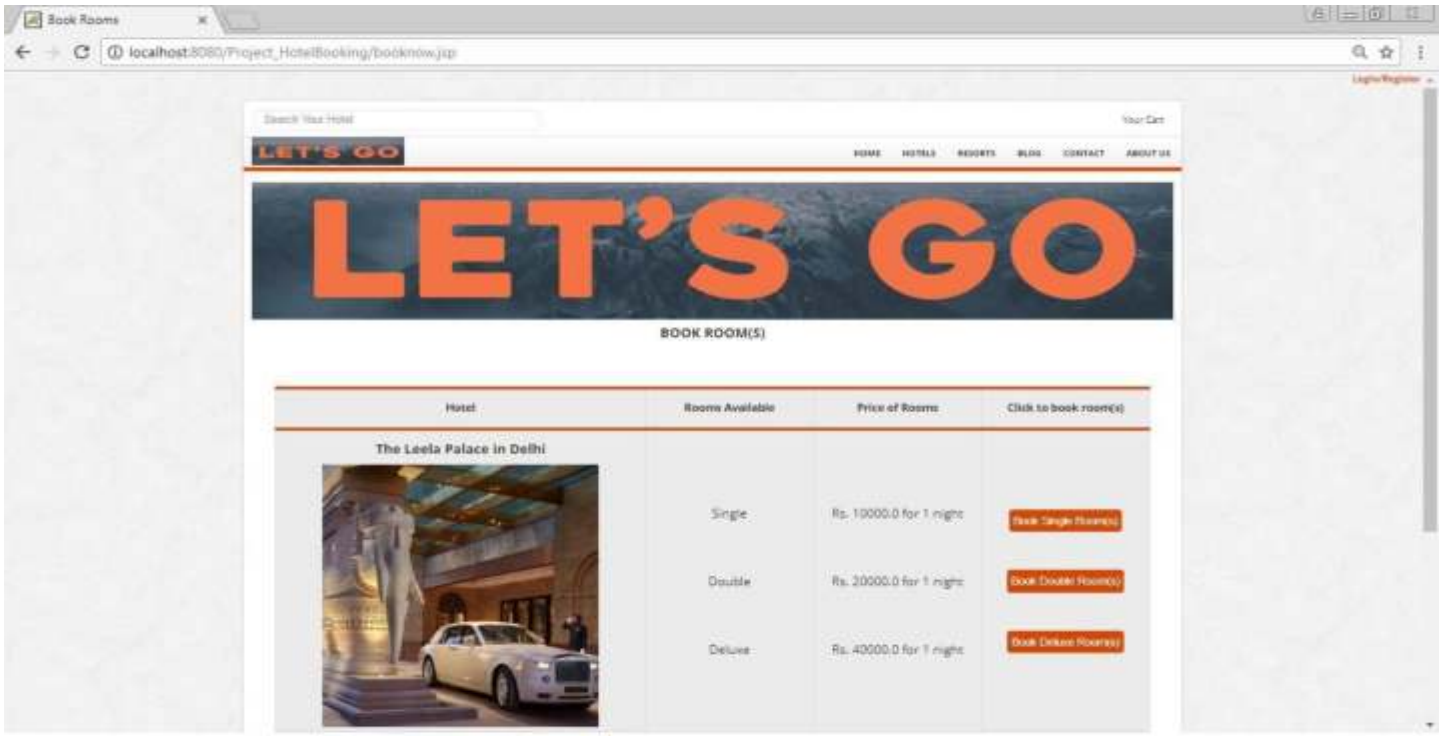
5) Clicking on the **'About Us'** menu takes the user to **'aboutus.jsp' page** which shows about the website and about its working principles.



6) After selecting a particular place on the **'hotels.jsp ' page** if user clicks on the '**Proceed**' button beside that selected place he/she will be directed to that **place's page** where all the hotels in that place will be displayed. For example, if a user wants to book a **hotel in 'Delhi'** then he/she has to click on **'Proceed' button in 'hotels.jsp' page** to see all the hotels in Delhi. As soon as he/she clicks on 'Proceed' button he/she is directed to **'Delhi.jsp' page** where **all the hotels in Delhi is displayed.** The same thing goes for other places as well such as Mumbai, Jaipur, Kochi etc.

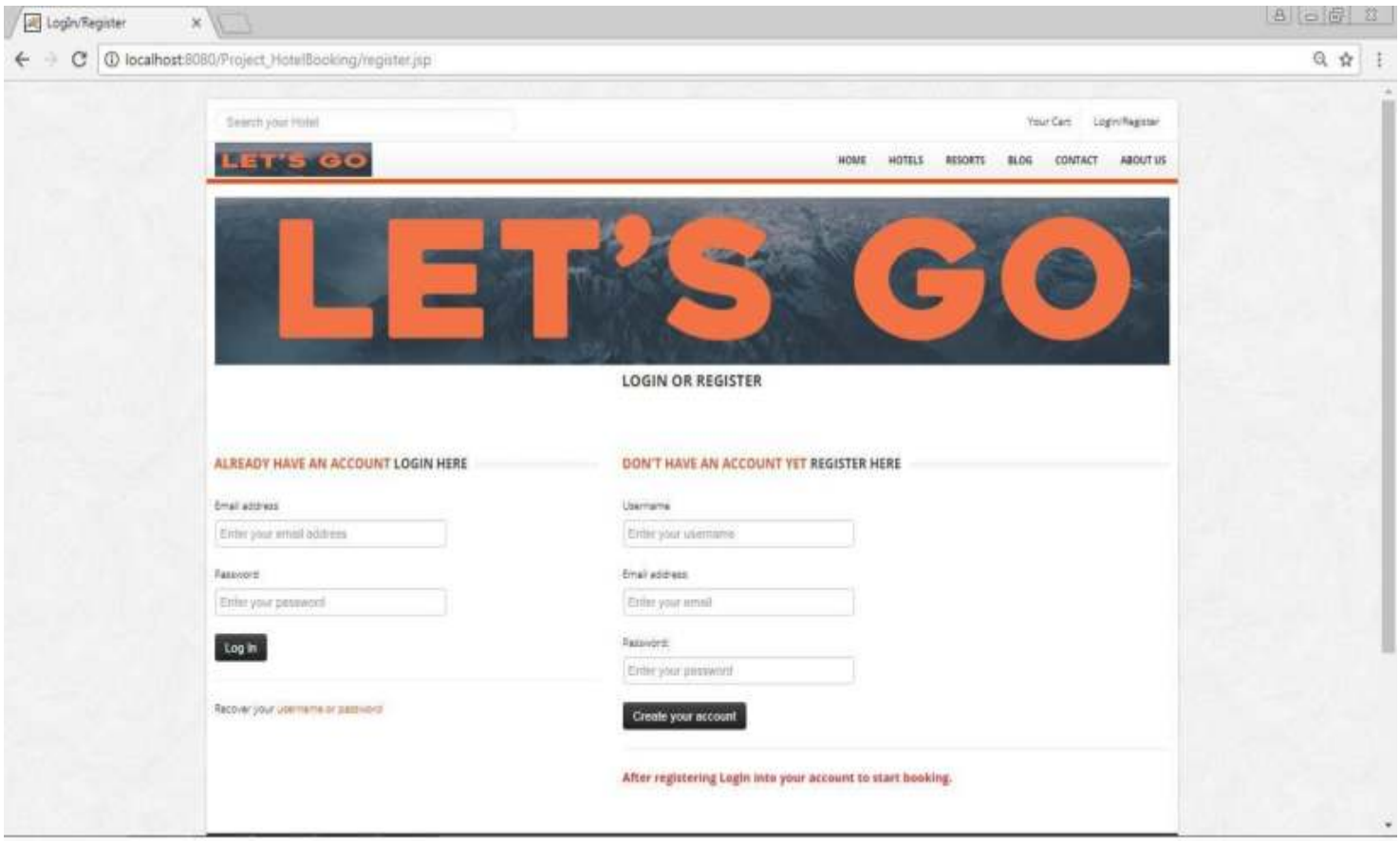


7) After selecting a particular hotel from the list of hotels the user needs to click on the **'Book Now'** button beside it to book a room in that hotel. Clicking on the 'Book Now' button directs the user to **'booknow.jsp' page** where the type of rooms present in the hotel and its prices are displayed. The user can now book a room depending on his/her budget & convenience.



8) If the user wants to book a single room in the hotel that he/she has selected then he/she must click on the **'Book Single Room(s)'** button. Similarly, for a double room on **'Book Double Room(s)'** button and for a deluxe room on **'Book Deluxe Room(s)'** button. Now, the user can only book a room if he/she is logged in into the website. If the user is not logged in then he/she will be automatically directed to the **'register.jsp' page** where the user has to login first into the website and then start booking**. Logged users are only allowed to book.** Unregistered users have to register first and then have to login into the website to start booking.

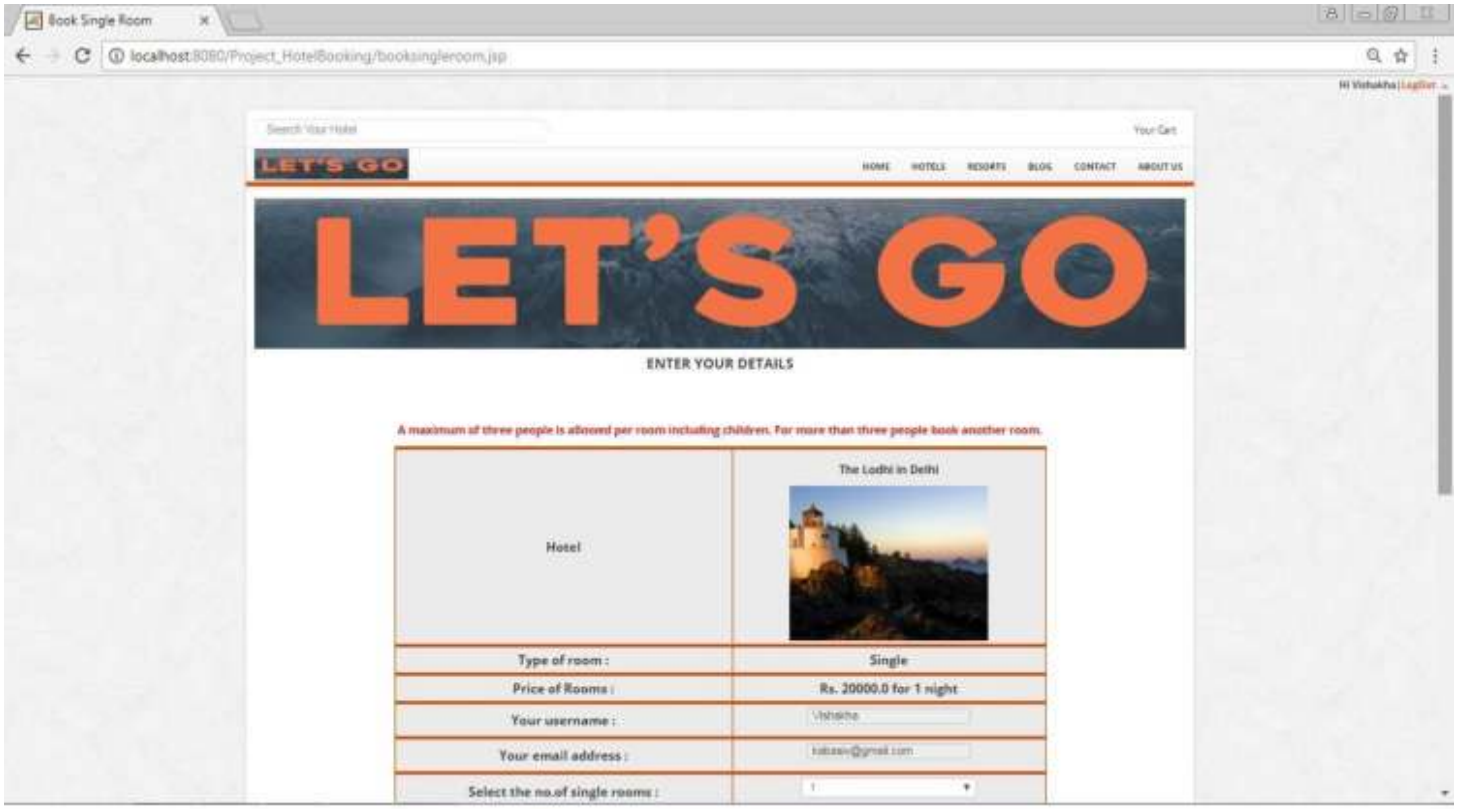
Here we consider that the user is not logged in first, so we can see that he/she is directed to the 'register.jsp' page as soon as he/she clicks on the button.

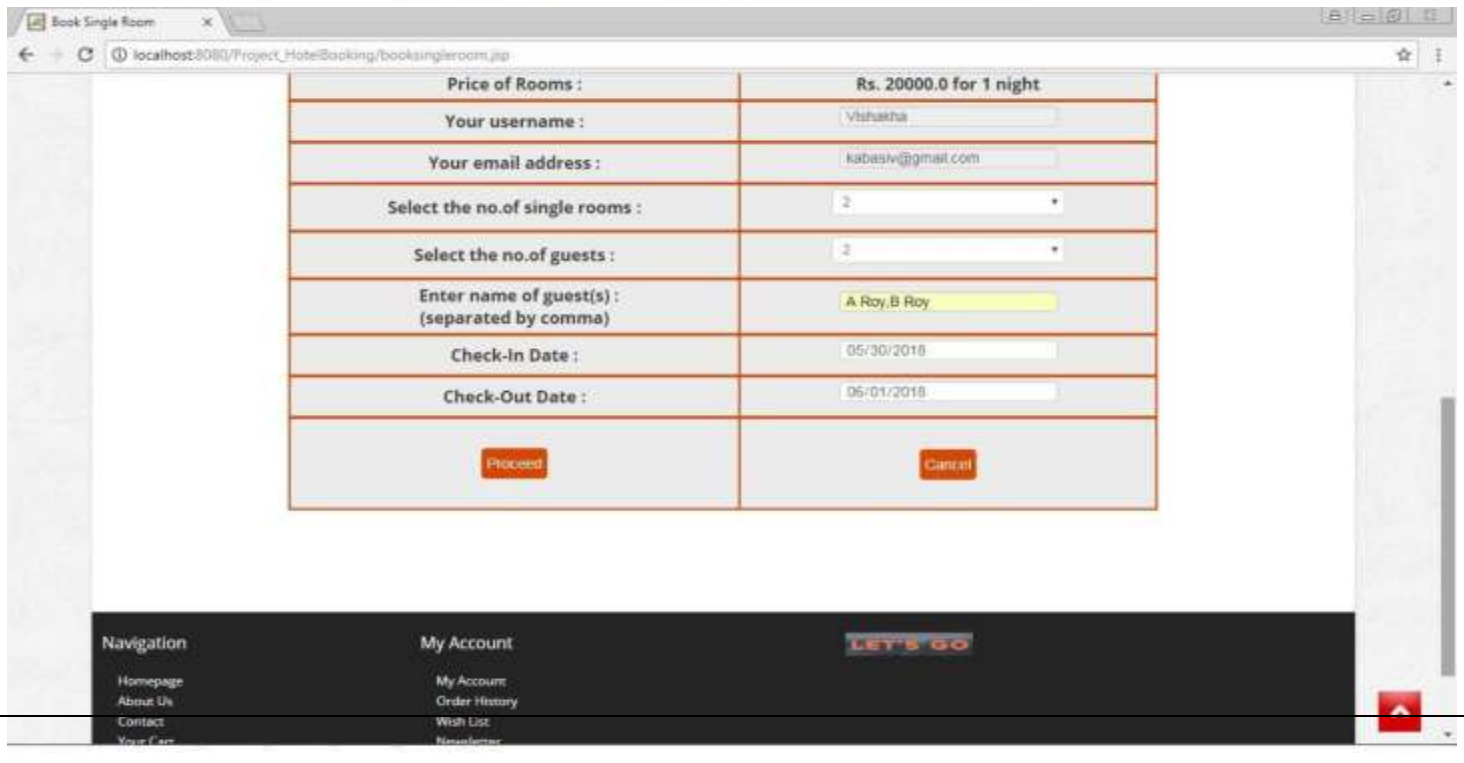


9) After logging into the website if the user clicks on 'Book Single Room(s)' button he/she is redirected to **'booksingleroom.jsp'** page where all the details of the room is displayed and the user has to enter his/her details required for booking the room.

Similarly for 'Book Double Room(s)' button goes to **'bookdoubleroom.jsp'** page and for 'Book Deluxe Room(s)' goes to '**bookdeluxeroom.jsp'** page.

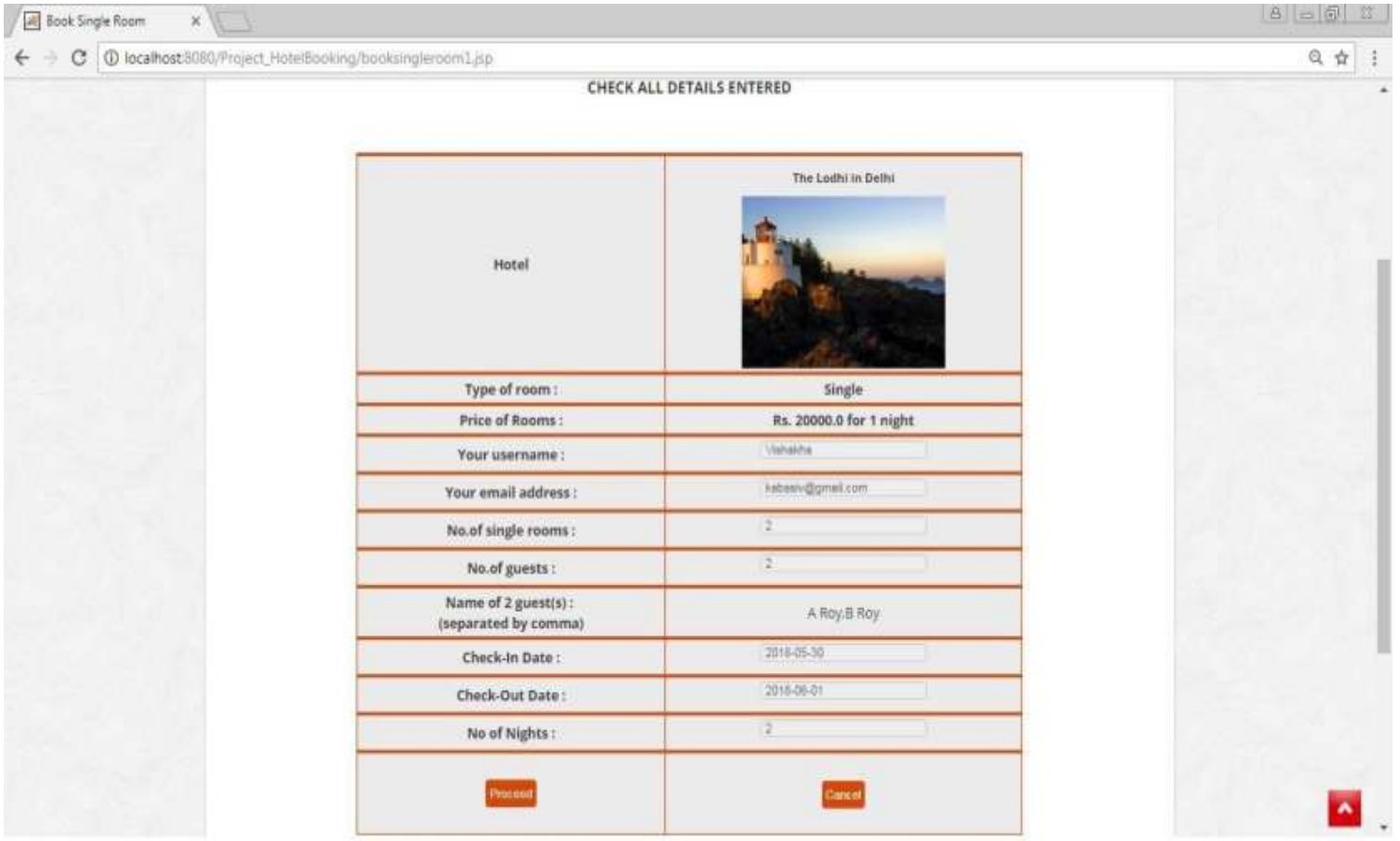
Here we have considered that the user is trying to book a single room.





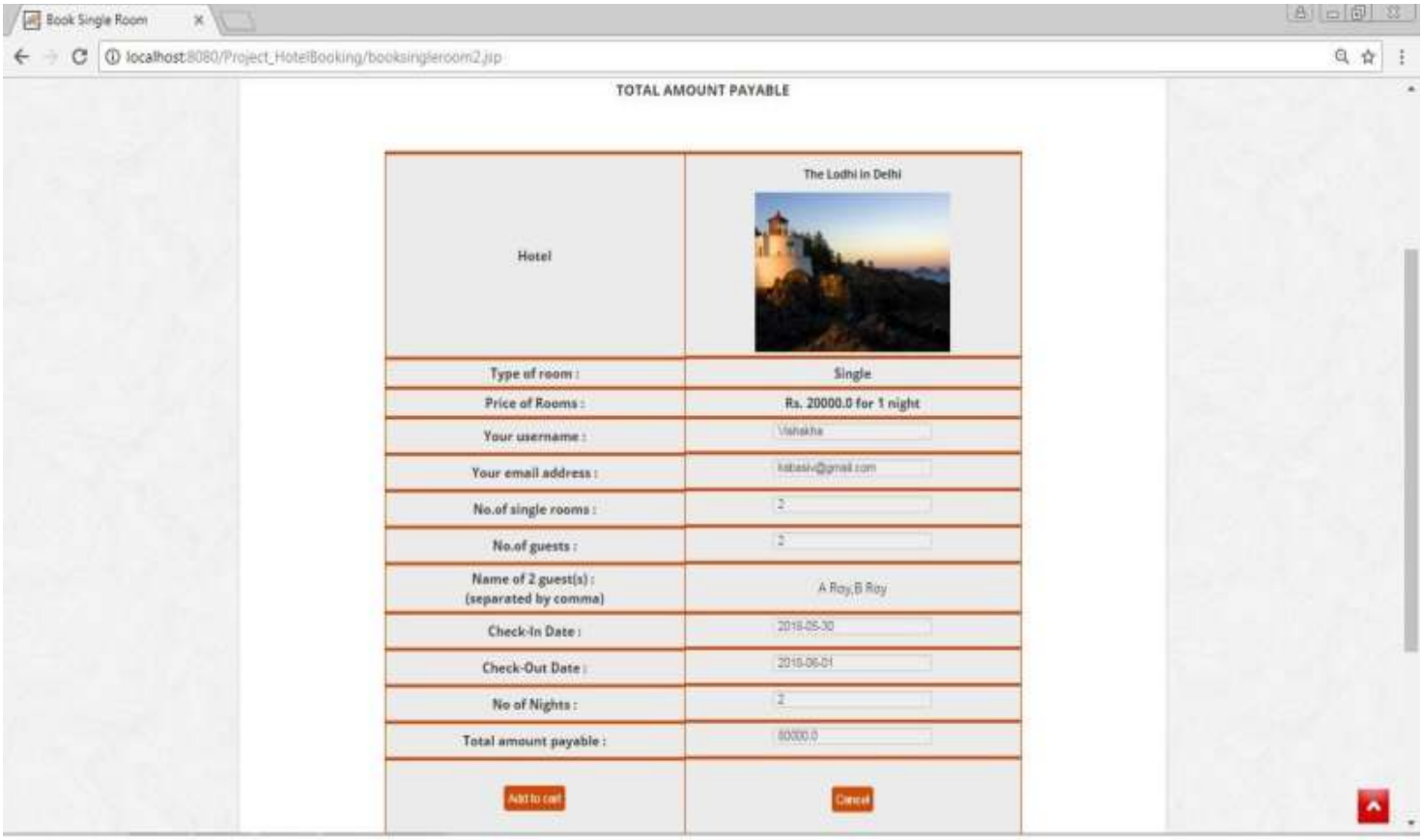
10) After clicking on the 'Proceed' button the user is directed to **'booksingleroom1.jsp'** page or **'bookdoubleroom1.jsp'** page or **'bookdeluxeroom1.jsp'** page depending on the button which he/she has clicked on. Here all the details that have been entered by the user is displayed.

Here we have considered that **the user is trying to book a single room.**

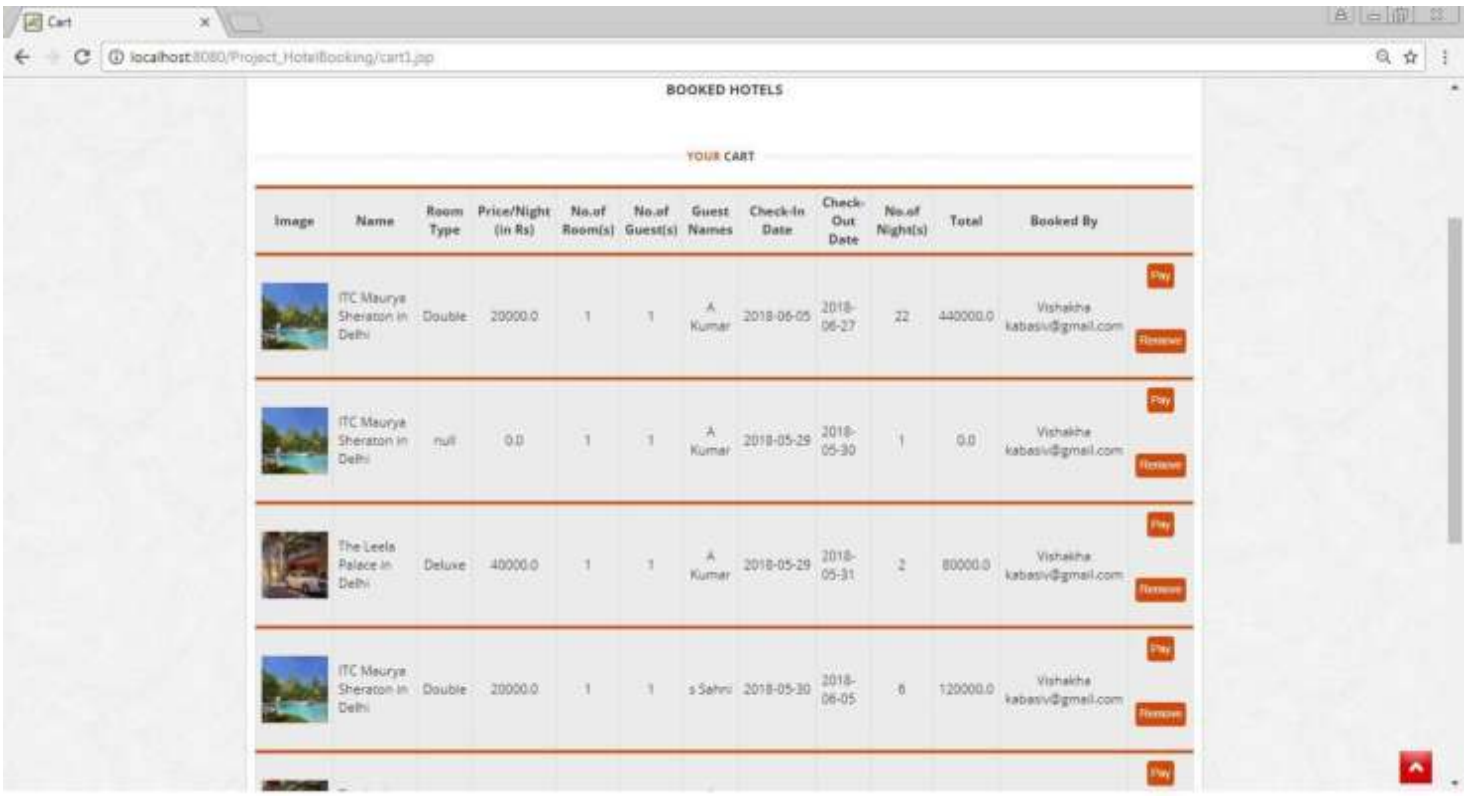


11) After clicking on **'Proceed' button** the user is directed to **'booksingleroom2.jsp' page** since we have considered that the user is trying to book a single room.

For Double room he/she would have been directed to **'bookdoubleroom2.jsp' page** and for Deluxe room he/she would have been directed to **'bookdeluxeroom2.jsp' page.** The amount to be paid by the users will be displayed in these directed pages.

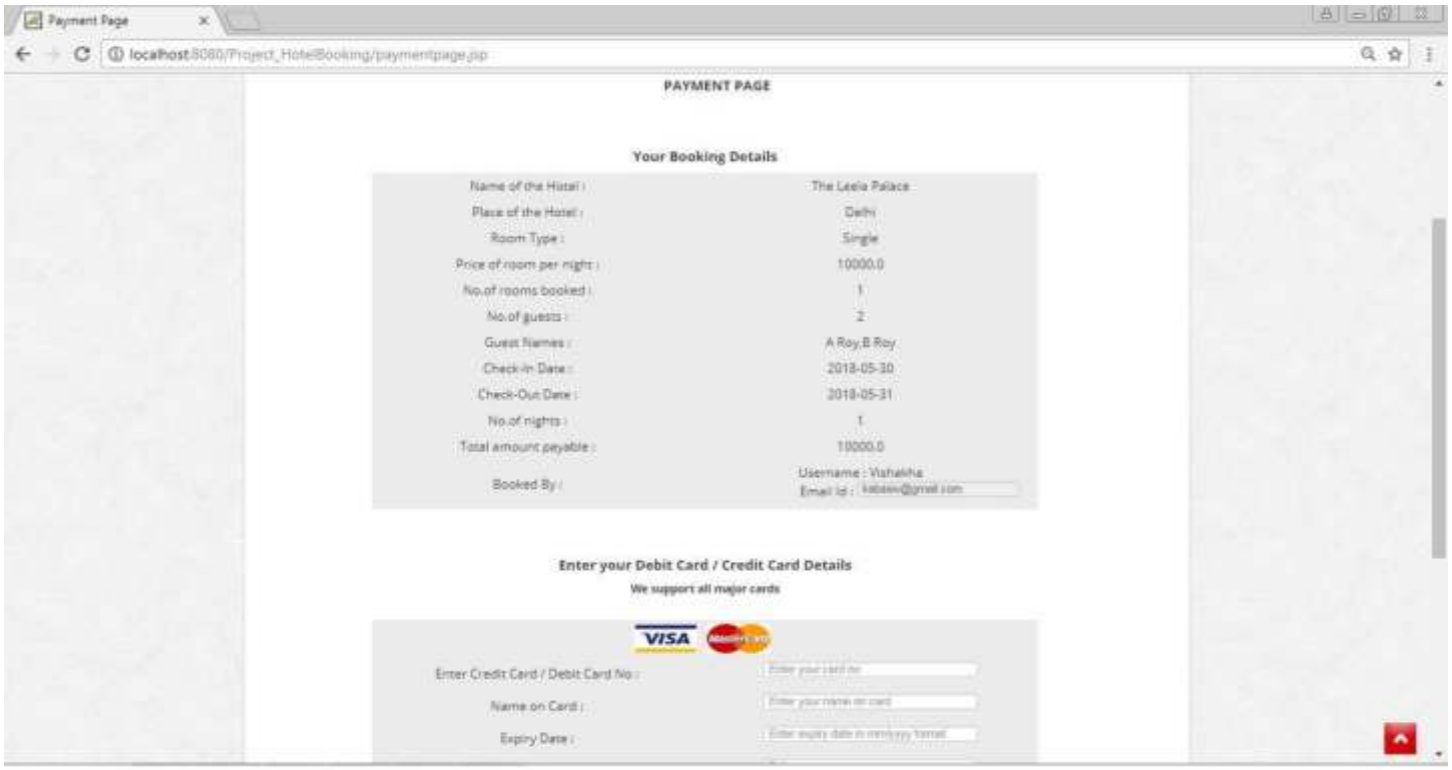


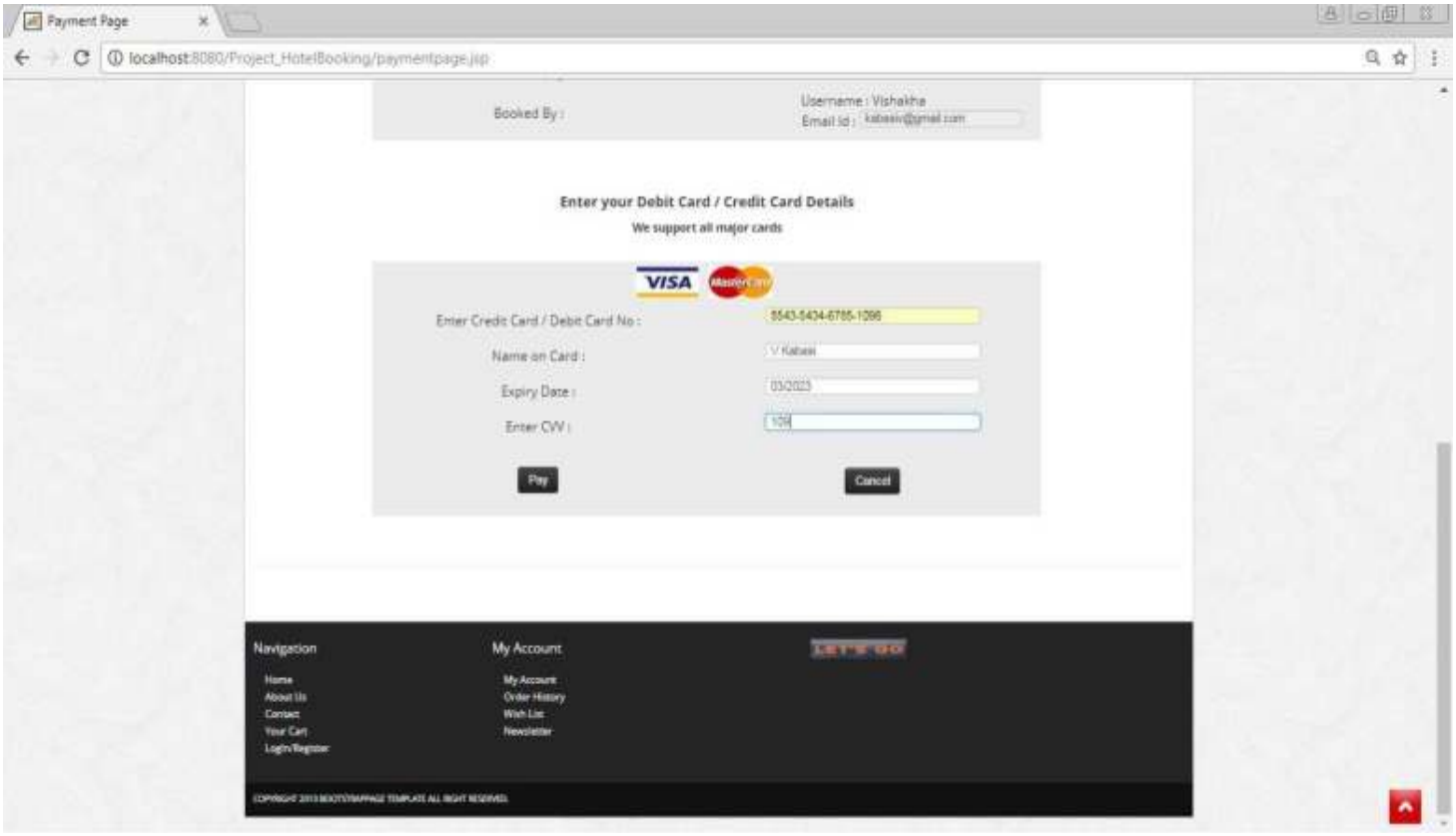
12) If the user clicks on **'Add to cart' button** the booked room gets added to the cart of the user and all the rooms that have been booked by the user is visible in the cart of the user. After clicking on 'Add to cart' button the user is directed to **'cart1.jsp' page** where the cart of the user is displayed.



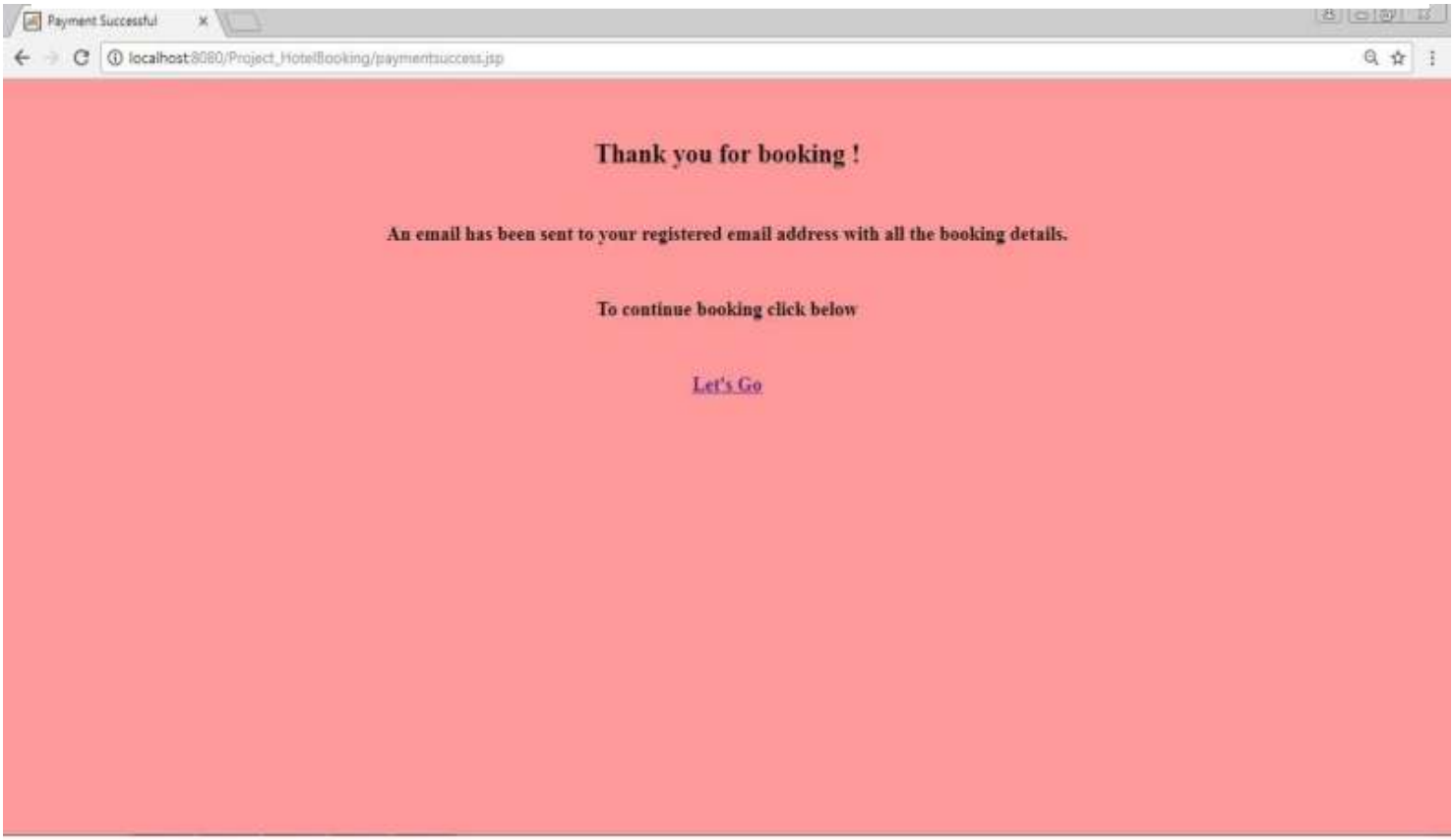
13) The user has two options now. He/She can decide to proceed further and **Pay** for the booked room or he/she can decide to **Remove/Cancel** the booked room from his/her cart.

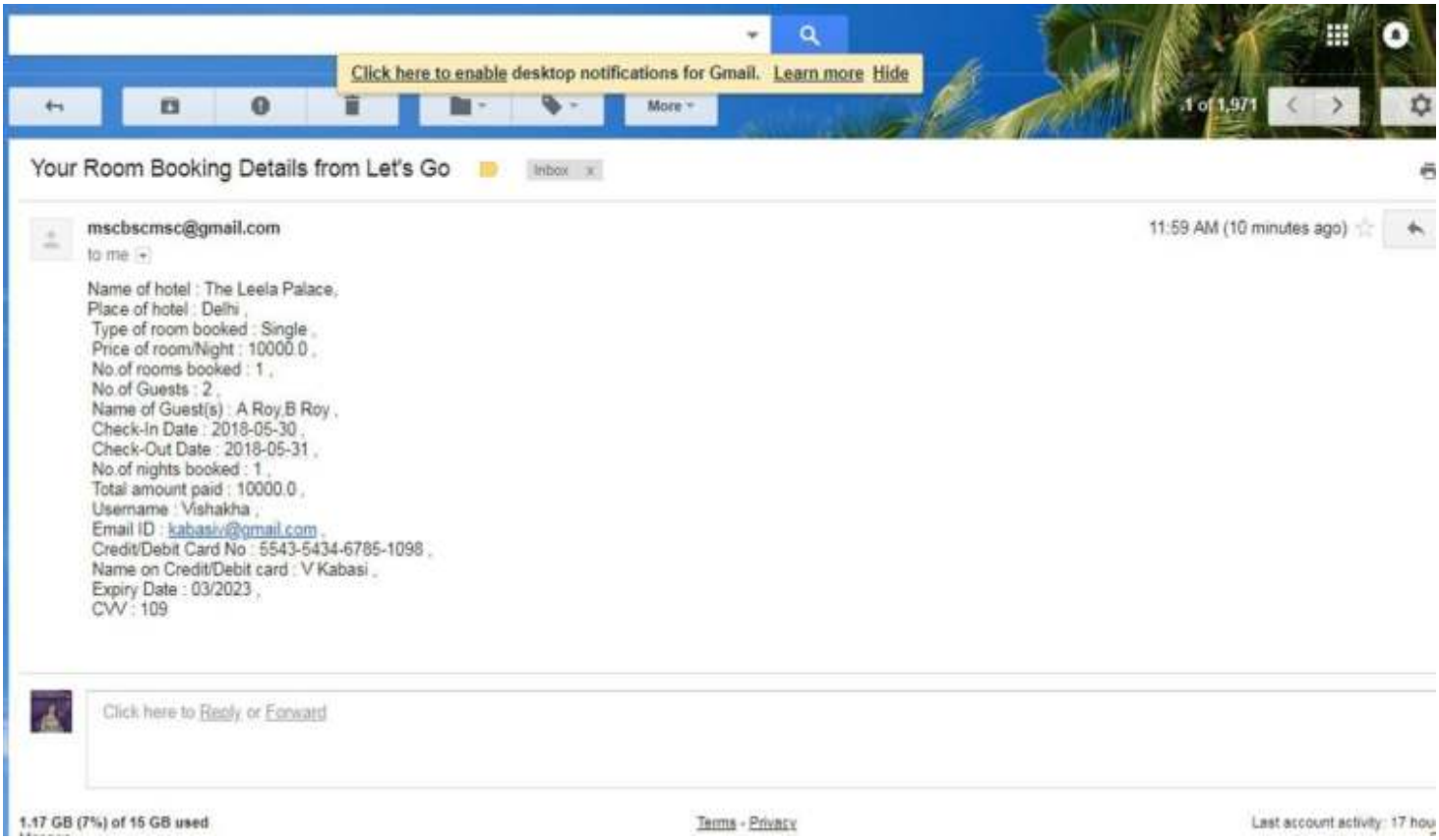
If the user clicks on **'Pay'** Button he/she is directed to **'paymentpage.jsp' page** where all the booking details entered by the user for that particular hotel room is displayed and the user now has to pay for that booked room via his/her credit/debit card to completely book the room. Clicking on the **'Remove'** button in the 'cart1.jsp' **page completely deletes the record of the booked room from his/her cart** and the user has to book it again from the beginning if the user wishes to do so.





14) After entering all the payment details if the user clicks on **'Pay' Button on the 'paymentpage.jsp' page** an **email is sent to the user** from Let's Go with all the booking details that have been entered by the user and the user is directed to **'paymentsuccess.jsp' page**.




Click here to enable desktop notifications for Gmail. Learn more Hide
More
1 of 1,971
Your Room Booking Details from Let's Go
Inbox x
mscbscmsc@gmail.com
to me
11:59 AM (10 minutes ago)
Name of hotel : The Leela Palace,
Place of hotel : Delhi ,
Type of room booked : Single ,
Price of room/Night : 10000.0 ,
No.of rooms booked : 1 ,
No.of Guests : 2 ,
Name of Guest(s) : A Roy,B Roy ,
Check-In Date : 2018-05-30 ,
Check-Out Date : 2018-05-31 ,
No.of nights booked : 1 ,
Total amount paid : 10000.0 ,
Username : Vishakha ,
Email ID : kabasiv@gmail.com ,
Credit/Debit Card No : 5543-5434-6785-1098 ,
Name on Credit/Debit card : V Kabasi ,
Expiry Date : 03/2023 ,
CVV : 109
Click here to Reply or Forward
1.17 GB (7%) of 15 GB used
Terms - Privacy
Last account activity: 17 hou

CONCLUSION

The main objective of the application is to help computer science students understand the basics of Java, JavaScript, and HTML. By browsing through the application and looking at the code for each graphical interpretation, students should be able to easily understand the implementation. The following results have been achieved after the completing the system and relate back to the system's objective.

1. **Should allow computer science students to browse through the code and application:** This is achieved when users, i.e., computer science students, are able to run and install the application. When they run the application, they can browse through the implementation of different objects.

2. **Should allow users to browse through different hotels in different places :** This is achieved when the user first runs the application and is directed to a home/index page that has hotels available for all places. The user can browse and click on any place to view all the hotels for that particular place.

3. Should allow users to save booked rooms to the cart and also to view detailed information about a particular room. The users can add any number of booked rooms according to their choice.

LIMITATION

- Application did not disclose full information.
- Data collected in this field are not sufficient.
- The system also don't have the online banking (**MayBank2U, CIMB Clicks, PayPal**) for customers.

FUTURE SCOPE OF THE PROJECT

This project ONLINE HOTELS & RESORTS BOOKING SITE has the following future scopes :-

The following section discusses the work that will be implemented with future releases of the software.

1. **Detailed places:** Future work could involve adding more places which are more detailed and have additional hotels in that place.

2. **Watch/Wish List:** Work can add a watch list or wish list so that users can add a room to a list to watch for room prices to go down or to see when there is a sale on any of those rooms.

3. **Enhanced User Interface:** Work on enhancing the user interface by adding more user-interactive features.

4. **Recommended Hotels:** Add a bar that would display the most-recommended hotels which would depend on the number of times a hotel has been booked by any users.

5. **Payment Options:** Add different payment options, such as Visa, MasterCard, PayPal, etc., where a user can also save the card information for later checkouts.

BIBLIOGRAPHY

- **Fundamentals of Software Engineering :**
  - Author: Rajib Mall
  - Publication: PHI

- **An Integrated Approach to Software Engineering**
  - Authors: Pankaj Jalota
  - Publication: Narosa Publishing House

- **Database System Concepts :**
  - Authors: Silberschatz, Korth & Sudarshan
  - Publication: McGraw Hill

- **Fundamentals of Database systems**
  - Author:Elmasri Navathe.
  - Publication: PHI.

- **Java Server Programming**
  - Black Book
  - Publication:Kogent

~~END~~