**Mobile Comparator System**

## A PROJECT REPORT

## Submitted in partial fulfillment of the Requirements for the Degree of

**MASTERS OF COMPUTER APPLICATIONS**

**by**

**Abhishek Verma**

**(1802914001)**

**Under The Supervision Of**

**Dr. SANGEETA ARORA**

**ASSOCIATE PROFESSOR**

****

**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**

## KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206

## (JUNE 2021)

## 

# DECLARATION

I hereby declare that the work presented in this report entitled “Mobile Comparator System ", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of another Institute or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Abhishek Verma

Enroll. No. : 1802914001

Branch : MCA

**(Abhishek Verma)**

# CERTIFICATE

Certified that **Abhishek Verma (1802914001)** has carried out the project work having “Mobile Comparator System” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other Institute/Institution.

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

**Dr. Sangeeta Arora**

**Associate Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay Shrivastava**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Date:**

**MOBILE COMPARATOR SYSTEM**

**ABHISHEK VERMA**

# ABSTRACT

**“Mobile Comparator System”** is a JSP (JAVA) application which is based on APACHE TOMCAT 7.0.

These days we are using databases in which coding is required to enter the data. But this time we have an application which can do our work more simpler and easier i.e JSP (JAVA). This application is used to store a huge amount of data properly and consistently.

The objective of this application is to show that how a normal person who doesn’t even know programming can use this application easily, it is flexible like data can be deleted enter or updated easily.A software project means a lot of experience. I learned a lot through this project. This project has sharpened our concept APACHE TOMCAT 7.0.

This concept of APACHE TOMCAT 7.0 has now become a great role to play in today's technical world. These technologies will definitely take database systems far away .

Through this project I learnt so many things can be manage through this application like sales, marketing, commercials and many more things. The only drawback of JSP (JAVA) is that it is expensive but to do great work we have to use good technology as today data security is the best and the most essential thing and JSP (JAVA) contains that all. things which I also learnt make Data Flow Diagram, State Event Diagram and Activity Diagram with help of our project Mentors

# ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my Project supervisor, **Dr. Sangeeta Arora** for her guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Srivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Abhishek Verma**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| Certificate |  | i |
| Abstract |  | ii |
| Acknowledgements |  | iii |
| Table of Contents |  | v-vii |

**CHAPTER 1: Introduction : 5-7**

1.1 Project details

1.2 Purpose of the Project

**CHAPTER 2: Technical Feasibility : 14-23**

* 1. JSP (JAVA)
  2. Standard Objects

**CHAPTER3: Creating Objects,Validations,Triggers : 23-28**

* 1. Creating JSP (JAVA) Objects
  2. Creating a Validation Rule
  3. coding

**Output Screens : 29-34**

**References : 35**

## LIST OF FIGURES

**Fig No. Title**

Fig 4.1 Home page

Fig 4.2 Registration Form

Fig 4.3 Change Profile Form

Fig 4.4 Admin login Form

Fig 4.5 Authority Page

Fig 4.6 Delete Registration Form

Fig 4.7 Registration list Form

Fig 4.8 Add new product Form

Fig 4.9 Edit/delete product Form

Fig 4.10 product list page

Fig 4.11 Change password form

Fig 4.12 Comparison page

## CHAPTER 1

**INTRODUCTION**

## Project Details:

“**Mobile Comparator System**” is taking a concept from “**APACHE TOMCAT 7.0**” and JSP (JAVA)

As the JSP (JAVA) is APACHE TOMCAT 7.0 Application consists of a large, waist-level cabinet with number of clouds in it. Each cloud contains different objects through which we can access the data. Once the application start, we can enter the particular amount of data which we need to put in the application for sure, the individual can directly add the data to the particular objects and fields thereby adding the students data. This do not require much time only we have to filled and enter all the data related to particular student.

This application will be developed for PC’s , we can access it from anywhere and even from the phone easily using the JSP (JAVA) application.

## Purpose:

JSP (JAVA) provides you with the fastest path from Idea to App. You can concentrate on building your app using JSP (JAVA) tools, rather than building the infrastructure and tools yourself. This can save you years of time and millions of dollars.

JSP (JAVA) customers generally say that it’s unique for three major reasons:

**Fast** – Traditional CRM software can take more than a year to deploy, compare that to months or even weeks with JSP (JAVA).

**Easy** – JSP (JAVA) wins in the easy to use category hands down. You can spend more time putting it to use and less time figuring it out.

**Effective** – Because it is easy to use and can be customized to meet business needs, customers find JSP (JAVA) very effective.

JSP (JAVA) is in the cloud, so your team can use it from anywhere with access to the internet. If you are a business that is rapidly changing or you are a seasoned company that’s been around for years, your business is probably changing too. JSP (JAVA) is completely scalable to your growth.

JSP (JAVA) seamlessly integrates with 3rd party apps. If you want to integrate JSP (JAVA) with Gmail you can do it, if you want to integrate it with your accounting software you can do that too. On the other hand, integration is tough with other CRMs. JSP (JAVA) is affordable, especially if you consider its vast variety of capabilities. Even startups and small businesses can use JSP (JAVA).

## Scope:

Students can use their device (laptop, mobile, desktop) or you can provide your lab.

We set up a branded solution for you (Software as a Service) and maintain every upgrade of the platform, right from hardware to software.

Our flagship product is built from scratch for engineering institutes with one objective – make students successful. Whether it is a campus job or higher education, we enable students and teachers to be successful by capturing, measuring, and analyzing the right data.

Engineering Students are at the core of the platform. We not only assess but also are part of the journey to accomplish the mission. Whether it is about finding a campus job or preparing for higher education, we enable you for success.

Students can learn from industry experts who are mentors on our platform (many of them are your college alumni). Take timely suggestions, join webinars and learn from their experiences. It is a virtual, ongoing, unstructured learning under supervised guidance that prepares you for the job ahead.

## CHAPTER 2

**APACHE TOMCAT 7.0**

## About:

APACHE TOMCAT 7.0 relies on software for distributed batch and stream processing, as well as distributed storage. This chapter focuses on an oft*-*ignored angle of assuredness: performance assuredness. A significant pain point today is the inability to support reconfiguration operations, such as changing of the shard key in a sharded storage/database system, or scaling up (or down) of the number of virtual machines (VMs) being used in a stream or batch processing system. We discuss new techniques to support such reconfiguration operations in an online manner, whereby the system does not need to be shut down and the user/client*-*perceived behavior is indistinguishable regardless of whether a reconfiguration is occurring in the background, that is, the performance continues to be assured in spite of ongoing background reconfiguration. Next, we describe how to scale*-*out and scale*-*in (increase or decrease) the number of machines/VMs in APACHE TOMCAT 7.0 frameworks like distributed stream processing and distributed graph processing systems, again while offering assured performance to the customer in spite of the reconfigurations occurring in the background. The ultimate performance assuredness is the ability to support SLAs/SLOs (service*-*level agreements/objectives) such as deadlines. We present a new real*-*time scheduler that supports priorities and hard deadlines for Hadoop jobs. We implemented our reconfiguration systems as patches to several popular and open*-*source APACHE TOMCAT 7.0 systems, including MongoDB and Cassandra (storage), Storm (stream processing), LFGraph (graph processing), and Hadoop (batch processing).[1]

APACHE TOMCAT 7.0 has become the industry standard for rapid application deployment, scalable server support, mobile and distributed services, and it provides access to (theoretically) infinite resources. Unfortunately, researchers are still trying to converge towards cross-provider APACHE TOMCAT 7.0 frameworks to enable compatibility and seamless resource transition between cloud providers. Moreover, users are restricted to using the provider-specific pre-configured options of resources and services, irrespective of their current needs. At the same time, cloud services are provided as a direct service from the providers to the clients. This creates a segregated cloud market clientele, and non-negotiable pricing

strategies for the cloud services. In this paper, we propose Jugo, a generic architecture for cloud composition and negotiated service delivery for cloud users. Jugo acts as a match-maker for service specifications from the users with the currently available assets from the cloud providers. The engagement of a middle-man as an opaque cloud service provider will create a better opportunity for cloud users to find cheaper deals, price-matching, and flexible resource specifications, with increased revenue and higher resource utilization for the cloud service providers.[2]

Many enterprises in industries start using APACHE TOMCAT 7.0 for their IT infrastructure services. This adoption of APACHE TOMCAT 7.0 is a part of the enterprise transformation which is the migration from a legacy IT environment to APACHE TOMCAT 7.0. On the other hand, one of major targets is an industry solution which provides a critical business service to their end customers. This paper proposes Industry Cloud which is the enhanced design of APACHE TOMCAT 7.0 for industry solutions. It efficiently supports industry solutions for enterprise business requirements. The paper describes Industry Cloud with a requirement analysis of industry solutions, those adopted functions, and three use case scenarios in the electronics and retail industry. The contribution of the paper is the analysis of industry wide requirements, the definition of Industry Cloud with a common function among industry solutions and the usage with use case scenarios.[3]

The complexity of Cloud infrastructures is increasing every year, requiring new concepts and tools to face off topics such as process configuration and reconfiguration, automatic scaling, elastic computing and healthiness control. This paper presents a Smart Cloud solution based on a Knowledge Base, KB, with the aim of modeling cloud resources, Service Level Agreements and their evolution, while enabling the reasoning on cloud structures and implementing strategies of efficient smart cloud management and intelligence. The solution proposed is composed of Smart Cloud Engine, SCE, the Knowledge Base, KB, and the Supervisor and Monitoring module for data acquisition. It can be easily integrated with any cloud configuration manager, cloud orchestra or, and monitoring tool, since the connections with these tools are performed by using REST calls and XML files.[4]

Current APACHE TOMCAT 7.0 is primarily based on proprietary data centers, where hundreds of thousands of dedicated servers are setup to host the cloud services. In addition to the huge number of dedicated servers deployed in data centers, there are billions of underutilized Personal Computers (PCs), usually used only for a few hours per day, owned by individuals and organizations worldwide. The vast untapped compute and storage capacities of the underutilized PCs can be consolidated as alternative cloud fabrics to provision broad cloud services, primarily infrastructure as a service. This approach, thus referred to as "no data center" approach, complements the data center based cloud provision model. In this paper, we present our opportunistic APACHE TOMCAT 7.0 system, called cuCloud, that runs on scavenged resources of underutilized PCs within an organization/community. Our system demonstrates that the "no data center" solution indeed works. Besides proving our concept, model, and philosophy, our experimental results are highly encouraging.[5]

Whatever one public cloud, private cloud or a mixed cloud, the users lack of effective security quantifiable evaluation methods to grasp the security situation of its own information infrastructure on the whole. This paper provides a quantifiable security evaluation system for different clouds that can be accessed by consistent API. The evaluation system includes security scanning engine, security recovery engine, security quantifiable evaluation model, visual display module and etc. The security evaluation model composes of a set of evaluation elements corresponding different fields, such as computing, storage, network, maintenance, application security and etc. Each element is assigned a three tuple on vulnerabilities, score and repair method. The system adopts “One vote vetoed” mechanism for one field to count its score and adds up the summary as the total score, and to create one security view.

We implement the quantifiable evaluation for different cloud users based on our G-Cloud platform. It shows the dynamic security scanning score for one or multiple clouds with visual graphs and guided users to modify configuration, improve operation and repair vulnerabilities, so as to improve the security of their cloud resources.[6]

To move applications to the cloud is not only a technical decision but also a business-oriented decision, in which both business and technical factors (e.g. transformation effort,

multi-tenancy and auto-scaling enablement, scalability and extensibility) should be considered. However, existing approaches and tools do not support a consumable business oriented cloud transformation decision to select more suitable transformation solution with the right cloud delivery model, services type, affordable transformation effort and etc. In this paper, we introduce a practical three-step approach and a tool, CTA (Cloud Transformation Advisor) to enable decision makers to identify the most suitable cloud transformation solution to satisfy their business goals based on a well-structured cloud transformation knowledge base.[7]

**CHAPTER 3**

**JSP (JAVA)**

The JSP (JAVA) Platform stores data in relational tables. The records in these tables contain data for the structure of the platform itself as well as user created data. For example, the data about the configuration and settings of an account are already in-built as a relational table. But you can also create your own tables to store data specific to your business like the 'dispatch schedule' for a week assuming you are a courier company.

These relational tables are roughly referred to as API Objects or only objects in JSP (JAVA).

There are three kinds of JSP (JAVA) objects.

* Standard Objects − The objects already created for you by the JSP (JAVA) platform.
* Custom Objects − These are the objects created by you based on your business processes.
* External Objects − The objects which you create map to the data stored outside your organization.
  1. **Standard Objects**

These are the objects which already exist in the JSP (JAVA) platform to manage the configurations and settings of the environment. Once you log in to the JSP (JAVA) platform, you can see the available objects.

**Important Standard Objects**

In this section, we will discuss the important standard objects in JSP (JAVA). The following table lists down the objects −

|  |  |  |
| --- | --- | --- |
| **bject Name** | **Meaning** | **Usage** |
| Account | Represents an individual account, which is an organization or person involved in the business like customers,  competitors,  partners, etc. | Use this object to query and manage accounts in your organization. |
| Account History | Represents the history of changes to the values in the fields of an account. | Use this object to identify changesto an  account. |
| Case | Represents a case, which is a customer issue or problem. | Use the case object to manage casesfor your organization. |
| Contact | Represents a contact, which is an individual associated with anaccount. | This object is used to manage individuals who are associated with an Account in the organization. |

|  |  |  |
| --- | --- | --- |
| ser | Represents a user in the organization. | This object is used to query information about users and also helps to provide and modify the information concerning the users. |
| Asset | Represents an item of commercial value, such as a product sold by the company or a competitor that a customer has purchased and installed. | This object is used to track assets previously sold into customer accounts. With asset tracking, a client application can quickly determine which products were previously sold or are currently installed at a specific account. |
| Domain | Read-only object that represents a custom Web address assigned to a site in your organization. | This read-only object is used to object to query the domains that are associated with each website in your organization. |

As IT technology advanced, a new style of innovation emerged, in which a leading innovation company invites end-users to its open software service platform. With respect to this type of innovation, a lot of innovation studies were performed to understand the structure of the interaction among users and the platform provider from the perspective of network science. By concentrating only on the internal mechanisms among agents, the previous studies miss to consider innovation through collective intelligence. A platform provider plays an important role in the innovation. In this research, we investigate the structure of a service network with empirical data gathered from JSP (JAVA).com App Exchange and discuss the role of a platform provider in innovation through collective intelligence. Our results suggest that the platform provider led the innovation in the initial period and, then, third party developers became gradually innovation leaders. Our findings are expected to re-orient the research focus from internal mechanisms to the role of platform providers.[8]

With the appearance of distributed computing, associations are hoping to move their Customer Relationship Management (CRM) applications from an On-Premise environment or we can say local servers to an On-Demand environment that is on cloud server. On-Premise environment is when association has the framework and programming inside their system. In On-Demand environment, an outsider has the base and programming and charges the relationship in light of its participation. JSP (JAVA) is the principle On-Demand CRM thing.[9]

The advantages of cloud while supporting real-time service systems using the JSP (JAVA) platform. We build here a service management platform for the Polish Billiards and Snooker Association (PBSA), based on a real-time system located in a cloud. It allows PBSA managers to accomplish tasks in this system on-demand. And, it is deployed as a private cloud to grant an access only to the employees from the snooker organization.[10]

In a recent scenario, IT industries are growing with the help of proper Utilization of available resources. The IT giants like Microsoft, Infosys, IBM, Oracle, & TCS are switching from theirs on premises IT setups to the cloud. APACHE TOMCAT 7.0 is replacing the traditional model in which software applications installed on on-premise hardware, from desktop computers to server rooms, depending on the size of the business. The proposed work is about the cloud platform which is going to change all the traditional views of software, application, and product development Technologies. JSP (JAVA).com is one of the best cloud providers available in the recent scenario. There are number of reasons why IT industries are switching to the Cloud. And there are number of reasons why Industries have to think to adopt JSP (JAVA).com cloud. The proposed work is about to focus on important and common features of salsforce.com. These features are common for any developer to learn and use in to software, application and product development in JSP (JAVA).com. The goal of this proposed work is to show the available resources in the JSP (JAVA).com which are still new for the developers. This an approach to make people familiar with the JSP (JAVA).com cloud provider.[11]

Summary form only given as follows. The strongest predictors of new product success is understanding market requirements early in the new product development (NPD) process. A direct JSP (JAVA)

is one of the best sources of new product ideas and market information, although not fully leveraged in many firms. A recent study of 248 salespeople in nineteen high-tech firms indicates wide variation in NPD involvement across and within a firm's JSP (JAVA). The study revealed that at the organization level, the length of the NPD cycle was associated with JSP (JAVA)-initiated and NPD-headquarters initiated activities. In particular, the longer a firm's NPD cycle for product improvements, the lower the involvement the JSP (JAVA) has in headquarters initiated involvement activities. A significant relationship exists between a firm's new product cycle time and the level of involvement in JSP (JAVA)-initiated NPD predevelopment activities. At the salesperson level, several factors affect involvement in NPD activities. Most significant was the relationship between salesperson involvement and the distance between a salesperson's location and the NPD office site. The study shows that organizations can affect the degree of involvement that their JSP (JAVA) or individual salespeople have in early phases of NPD.[12]

Motivation is important because it determines and individual's effort toward performing a task and it leads to an optimistic and challenging attitude at work place. Incentive motivation is concerned with the way goals influence behavior. Incentive is the promise or stimulus for greater action. Incentives are something that is given in addition to wages. It means additional remuneration or benefit to an employee in recognition of achievement or better work. Organizations in emerging markets are more likely to give group rewards and generally do not encourage risk taking as much as developed economies. Money motivates employees to some extent but there are other powerful sources of motivation such as interesting and challenging tasks and assignments, praise, good interpersonal relations etc. This paper analyzes incentives plan structure of sales people in Croatia. Advantages and disadvantages of different compensation plans and benefits for salespeople are discussed.[13]

Modern business decision models are often very complicated due to a deluge of information. Evaluation and diagnostics of such decision models is extremely challenging due to many factors, including the complexity and volume of data. In addition, since there is no ideal data sample to construct a control group for comparison studies, performance evaluation and diagnostics of business actions can easily be distorted by selection bias. In this paper, we design a framework to analyze this sample bias issue under a practical business scenario. In particular, we focus on: a) identification of the key factors which drive selection bias during the business decision; b) evaluation of the performance of business actions with consideration of the identified selection bias. We evaluate baseline analytics tools on the worldwide sales- force data of a large global corporation and clearly demonstrate that the selection bias issue makes the

usual evaluation very unstable and not trustable. However, by removing such detected sample bias, our framework can generate reasonable diagnostics results across different dimensions. The implemented analysis tool was applied to a worldwide business opportunity dataset of a multinational Fortune 500 corporation; the analytics results clearly show the significance of such a bias detection-based evaluation framework for sales-force optimization.[14]

Estimating generalizable relationships between actions and results from historical samples, especially when there is a level of noise or randomness in that signal, is an important problem to address before making decisions on actions to take. Many business analytics problems require the optimal assignment of limited resources to actions and activities to maximize some result or objective such as profit. We present a novel approach to solving this class of analytics problems by modeling the relationship between resource effort and expected return as a dose-response signal and formulating its causal estimation as one of kernel regression. The estimated expected value and variance of the result are then used to optimize resource allocation so as to maximize expected response while minimizing the risk around response subject to business constraints. We apply this approach to the task of optimally assigning salespeople to enterprise clients using real-world data, and show that profit can be substantially increased with fewer salespeople and reduced risk.[15]

**CHAPTER 3.1**

* + 1. **Creating a JSP (JAVA) Custom Object**
* First, follow this path: Setup > Build > Create > Objects > on the screen, click on the new custom object button > enter label name, plural label, and object name > enter record name as a data type.
* To create a record name, first, consider the two data types:
* Text
* Auto-number

You also have some optional features while creating objects:

* Allow reports: If you tick off this checkbox, then only these objects will be available to make reports.
* Allow activities: If you tick off this checkbox, then you are able to make activities on this object.
* Track field history: When you tick off this checkbox, then you can merely track fields. You can track up to 20 fields for a single object.

Now, you have the following deployment modes:

* In development: If you opt for this, then this object will remain in the development mode. It will not be present for deployment.
* Deployed: When you opt for this, the object will be available for deployment.

The following options are available only when you are creating a custom object for the first time:

* Add notes and attachments related list
* Launch a new custom tab wizard after saving this custom object After completing all these settings, press the Save button.

If you do not select ‘Launch new custom tab wizard’ from the object creation page, then the object would be saved without the tab appearance. In such a case, first, you need to make a tab for this object. On other hand, if you select this option, the object would be saved, along with a tab would be created, which will be visible to you.

3.1.2

**COMPLETE PROJECT CODING**

**Home Page**

<html>

<body bgcolor=teal style="margin:0">

<img src="image/mobile3.jpg" width=10% height=15% style="float:left">

<div style="Background-color:powderblue; width:55%; height:12%; float:left; padding:10px">

<font color=crimson size=7>

<b> S M A R T R I X </b>

</font>

<br>

<font color= darkblue size=5>

<b> MOBILE PHONE COMPARATOR </b>

</font>

</div>

<div style="Background-color:powderblue; width:31.7%; height:12%; float:left; padding:10px">

<marquee behavior=alternate>

<img src="image/mobile1.jpg" width=95 height = 75>

<img src="image/mobile2.jpg" width=95 height = 75>

<img src="image/mobile3.jpg" width=95 height = 75>

</marquee>

</div>

<fieldset style="background-color:skyblue;border:none;width:100%; height:9.5%; margin-left:-5">

<center>

<br>

<a href=main.jsp target=frm> <input type=button value=HOME style="width:150;height:30;background-color:darkblue; color:white; border:none"> </a>

<a href=registration.jsp target=frm> <input type=button value=REGISTRATION style="width:150;height:30;background-color:darkblue; color:white; border:none"> </a>

<a href=changeprofile.jsp target=frm> <input type=button value="CHANGE PROFILE" style="width:150;height:30;background-color:darkblue; color:white; border:none"> </a>

<a href=admin.jsp> <input type=button value=ADMINSTRATOR style="width:150;height:30;background-color:darkblue; color:white; border:none"> </a>

<a href=comparison.jsp> <input type=button value=COMPARISON style="width:150;height:30;background-color:darkblue; color:white; border:none"> </a>

<input type=button value="CONTACT US" style="width:150;height:30;background-color:darkblue; color:white; border:none">

</fieldset>

<iframe name=frm width=100% height=68% src=main.jsp>

</iframe>

<div style="background-color:darkblue;width:100%;height:4%">

<center>

<font color=yellow size=4>

Copyright:www.smartrix.com

</font>

</center></div>

</body

</html>

**Registration Page**

<%@page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue text=darkblue>

<form>

<h2 style="color:crimson">

REGISTARTION FORM

</h2>

<hr color=blue size=1>

<table cellspacing=10>

<tr>

<td> Registraion No </td>

<td> <input type=text name=txtrno size=10> </td>

</tr>

<tr>

<td> Registraion Date </td>

<td> <input type=text name=txtrdate size=20> </td>

</tr>

<tr>

<td> Name </td>

<td> <input type=text name=txtname size=20> </td>

</tr>

<tr>

<td> Gender </td>

<td> <input type=radio name=gen value=Male> Male

<input type=radio name=gen value=Female checked> Female </td>

</tr>

<tr>

<td> Password </td>

<td> <input type=password name=txtpass size=20> </td>

</tr>

<tr>

<td> Confirm Password </td>

<td> <input type=password name=txtconfirm size=20> </td>

</tr>

<tr>

<td> Address </td>

<td colspan=3> <input type=text name=txtadd size=40> </td>

</tr>

<tr>

<td> Phone No </td>

<td> <input type=text name=txtphn size=20> </td>

</tr>

<tr>

<td> Email ID </td>

<td> <input type=email name=txtemail size=40> </td>

</tr>

</table>

<hr color=blue size=1>

<input type=submit name=b1 Value=SUBMIT style="background-color:darkblue; color:white; width:8%; height:25px">

<input type=reset name=b2 value=RESET style="background-color:darkblue; color:white; width:8%; height:25px">

<%

String r1=request.getParameter("b1");

if("SUBMIT".equals(r1))

{

int rno;

String rdate,name, gender, pass, confirm, address, phn, eml;

rno=Integer.parseInt(request.getParameter("txtrno"));

rdate=request.getParameter("txtrdate");

name=request.getParameter("txtname");

gender=request.getParameter("gen");

pass=request.getParameter("txtpass");

confirm=request.getParameter("txtconfirm");

address=request.getParameter("txtadd");

phn=request.getParameter("txtphn");

eml=request.getParameter("txtemail");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q1="insert into reg values("+rno+",'"+rdate+"','"+name+"','"+gender+"','"+pass+"','"+address+"','"+phn+"','"+eml+"')";

state.executeUpdate(q1);

out.write("<h2> Successfully registered </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</form>

</body>

</html>

**Change Profile Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue>

<form>

<center>

<h2 style="color:maroon">

<u> CHANGE PROFILE </u>

</h2>

<table cellspacing=10>

<tr>

<td> Enter Name </td>

<td> <input type=text name=t1 size=20> </td>

</tr>

<tr>

<td> Enter Password </td>

<td> <input type=password name=t2 size=20> </td>

</tr>

</table>

<input type=submit value=VERIFY name=b1 style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

String r1=request.getParameter("b1");

if("VERIFY".equals(r1))

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String nm=request.getParameter("t1");

String pw=request.getParameter("t2");

String q1="select \* from reg where name='"+nm+"' and password='"+pw+"'";

ResultSet res=state.executeQuery(q1);

if(!res.next())

out.write("Not Found");

else

{

%>

<hr>

<table cellspacing=10>

<tr>

<td> <input type=hidden name=t3 size=20 value="<%out.write(res.getString("name"));%>"> </td>

</tr>

<tr>

<td> <input type=password hidden name=t4 size=20 value="<%out.write(res.getString("password"));%>"> </td>

</tr>

<tr>

<td> Password </td>

<td> <input type=password name=t5 size=20 value="<%out.write(res.getString("password"));%>"> </td>

</tr>

<tr>

<td> Confirm Password </td>

<td> <input type=password name=t6 size=20> </td>

</tr>

<tr>

<td> Address </td>

<td colspan=3> <input type=text name=t7 size=20 value="<%out.write(res.getString("address"));%>"> </td>

</tr>

<tr>

<td> phone </td>

<td> <input type=text name=t8 size=20 value="<%out.write(res.getString("phone"));%>"> </td>

</tr>

<tr>

<td> email </td>

<td> <input type=email name=t9 size=20 value="<%out.write(res.getString("email"));%>"> </td>

</tr>

</table>

<br>

<input type=submit name=b2 value=UPDATE style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

String r2=request.getParameter("b2");

if("UPDATE".equals(r2))

{

String nm1=request.getParameter("t3");

String opw=request.getParameter("t4");

String npw=request.getParameter("t5");

String add=request.getParameter("t7");

String ph=request.getParameter("t8");

String em=request.getParameter("t9");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q3="update reg set password='"+npw+"',address='"+add+"',phone='"+ph+"',email='"+em+"' where name='"+nm1+"' and password='"+opw+"'";

state.executeUpdate(q3);

out.write("<h2> Successfully Updated </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</center>

</form>

</body>

</html>

**Admin Page**

%@ page import="java.sql.\*" %>

<html>

<body bgcolor=teal>

<form>

<Center>

<div style="background-color:lightblue; color:darkblue; width:35%; height:80%; margin-top:50px">

<center>

<br>

<h2 style="color:crimson"> ADMINSTRATOR LOGIN </h2>

<img src="image/reg.jpg" width=40% height=30%>

<p>

<br>

<input type=text name=t1 size=30 placeholder="<NAME>" >

<p>

<input type=password name=t2 size=30 placeholder= "<PASSWORD>" >

<p>

<input type=submit name=blgn value=LOGIN style="background-color:darkblue; color:white;">

<%

String r=request.getParameter("blgn");

if("LOGIN".equals(r))

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String name=request.getParameter("t1");

String pass=request.getParameter("t2");

String q="select \* from admin where name='"+name+"' and password='"+pass+"'";

ResultSet rs=state.executeQuery(q);

if(!rs.next())

out.write(" <h2> Incorrect Password </h2> ");

else

{

response.sendRedirect("authority.jsp");

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</center>

</div>

</center>

</form>

</body>

</html>

**Authority Page**

<html>

<body bgcolor=teal style="margin:0">

<img src="image/mobile3.jpg" width=10% height=15% style="float:left">

<div style="Background-color:skyblue; width:55%; height:12%; float:left; padding:10px">

<font color=crimson size=7>

<b> S M A R T R I X </b>

</font>

<br>

<font color= darkblue size=5>

<b> MOBILE PHONE COMPARATOR </b>

</font>

</div>

<div style="Background-color:skyblue; width:31.7%; height:12%; float:left; padding:10px">

<marquee behavior=alternate>

<img src="image/mobile1.jpg" width=95 height = 75>

<img src="image/mobile2.jpg" width=95 height = 75>

<img src="image/mobile3.jpg" width=95 height = 75>

</marquee>

</div>

<br>

<div style="background-color:powderblue; width:25%; height:85%; float:left">

<center>

<h3 style="color:crimson"> <u> ADMINSTRATOR AUTHORIES </u> </h3>

</center>

<table cellspacing=20>

<tr>

<td> <img src="image/rdelete.jpg" width=120 height =60>

<br>

<a href=rdelete.jsp target=frm1> <font color=darkblue size=4.5> Delete Reg </font> </a>

</td>

<td> <img src="image/plist.png" width=120 height = 60>

<br>

<a href=reglist.jsp target=frm1> <font color=darkblue size=4.5> Reg List </font> </a>

</td>

</tr>

<tr>

<td> <img src="image/padd.jpg" width=120 height =60>

<br>

<a href=addproduct.jsp target=frm1> <font color=darkblue size=4.5> Add Product </font> </a>

</td>

<td> <img src="image/pdelete.jpg" width=120 height = 60>

<br>

<a href=editDelpro.jsp target=frm1> <font color=darkblue size=4.5> Edit/Delete Product </font> </a>

</td>

</tr>

<tr>

<td> <img src="image/plist.png" width=120 height =60>

<br>

<a href=prolist.jsp target=frm1> <font color=darkblue size=4.5> Product List </font> </a>

</td>

<td> <img src="image/change.jpg" width=120 height = 60>

<br>

<a href=change.jsp target=frm1> <font color=darkblue size=4.5> Change Password </font> </a>

</td>

</tr>

<tr>

<td> <img src="image/logout.png" width=120 height =60>

<br>

<a href=logout.jsp target=frm1> <font color=darkblue size=4.5> Logout </font> </a>

</td>

</tr>

</table>

</div>

<div style="background-color:powderblue; width:75%; height:92%;float:right">

<iframe name=frm1 width=100% height=90%>

</iframe>

</div>

</body>

</html>

**Delete Registration Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue>

<form>

<h2> <font color=darkblue> DELETE REGISTRATION </font> </h2>

<hr>

<font color=blue size=4>

Enter Reg No <input type=text name=t1 size=20>

</font>

<br>

<br>

<input type=submit name=b1 value=SEARCH style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

String r1=request.getParameter("b1");

if("SEARCH".equals(r1))

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

int rn=Integer.parseInt(request.getParameter("t1"));

String q1="select \* from reg where rno="+rn;

ResultSet res=state.executeQuery(q1);

if(!res.next())

out.write("Not Found");

else

{

%>

<hr>

<table cellspacing=10>

<tr>

<td> Reg No </td>

<td> <input type=text disabled name=t2 size=20 value=<% out.write(res.getString("rno")); %>> </td>

</tr>

<tr>

<td> Reg Date </td>

<td> <input type=text name=t3 size=20 value=<% out.write(res.getString("rdate")); %>> </td>

</tr>

<tr>

<td> Name </td>

<td> <input type=text name=t4 size=20 value=<% out.write(res.getString("Name")); %>> </td>

</tr>

<tr>

<td> Gender </td>

<td> <input type=text name=gen size=20 value=<% out.write(res.getString("gender")); %>>

</tr>

<tr>

<td> Password </td>

<td> <input type=passowrd name=t5 size=20 value=<% out.write(res.getString("password")); %>> </td>

</tr>

<tr>

<td> Address </td>

<td colspan=3> <input type=text name=t6 size=20 value=<% out.write(res.getString("address")); %>> </td>

</tr>

<tr>

<td> Phone no </td>

<td> <input type=text name=t7 size=20 value=<% out.write(res.getString("phone")); %>> </td>

</tr>

<tr>

<td> Email </td>

<td> <input type=email name=t8 size=20 value=<% out.write(res.getString("email")); %>> </td>

</tr>

</table>

<hr>

<br>

<input type=submit name=b2 value=DELETE style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

String r2=request.getParameter("b2");

if("DELETE".equals(r2))

{

int n1=Integer.parseInt(request.getParameter("t2"));

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q2="delete from reg where rno="+n1;

state.executeUpdate(q2);

out.write("<h2> Successfully Deleted </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</form>

</body>

</html>

**Change Password Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue>

<form>

<h2> <font color=darkblue> CHANGE PASSWORD </font> </h2>

<hr>

<font color=darkblue size=4>

Enter Current Password <input type=password name=t1 size=20>

</font>

<br>

<br>

<input type=submit name=b1 value=VERIFY style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

String r1=request.getParameter("b1");

if("VERIFY".equals(r1))

{

String p= request.getParameter("t1");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q1="select \* from admin where password='"+p+"' ";

ResultSet res=state.executeQuery(q1);

if(!res.next())

out.write(" <h2> Incorrect </h2> ");

else

{

%>

<hr>

<table cellspacing=10>

<tr>

<td> <font color=darkblue size=4> Enter New Password </font> </td>

<td> <input type=password name=t2 size=20> </td>

</tr>

<tr>

<td> <font color=darkblue size=4> Confirm Password </font> </td>

<td> <input type=password name=t3 size=20> </td>

</tr>

</table>

<hr>

<input type=submit name=b2 value=UPDATE style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

String r2=request.getParameter("b2");

if("UPDATE".equals(r2))

{

String newpass,confirm;

newpass=request.getParameter("t2");

confirm=request.getParameter("t3");

if(!newpass.equals(confirm))

out.write("<h2>Not match</h2>");

else

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q2="update admin set password='"+request.getParameter("t2")+"'";

state.executeUpdate(q2);

out.write("<h2> Successfully Updated </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

}

%>

</form>

</body>

</html>

**Registration List Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue text=darkblue>

<center>

<h2 style="color:red">

<u> REGISTRATION LIST </u>

</h2>

<table width=1000 border=2 cellpadding=10>

<tr>

<td> <h2> Reg No </h2> </td>

<td> <h2> Reg Date </h2> </td>

<td> <h2> Name </h2> </td>

<td> <h2> Gender </h2> </td>

<td> <h2> Address </h2> </td>

<td> <h2> Phone No </h2> </td>

<td> <h2> Email </h2> </td>

</tr>

<%

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q= "select \* from reg";

ResultSet res= state.executeQuery(q);

while(res.next())

{

%>

<tr>

<td> <% out.write(res.getString("Rno")); %> </td>

<td> <% out.write(res.getString("Rdate")); %> </td>

<td> <% out.write(res.getString("name")); %> </td>

<td> <% out.write(res.getString("Gender")); %> </td>

<td> <% out.write(res.getString("Address")); %> </td>

<td> <% out.write(res.getString("phone")); %> </td>

<td> <% out.write(res.getString("Email")); %> </td>

</tr>

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

%>

</table>

</center>

</body>

</html>

**Add Product Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue text=darkblue style="margin:0">

<form>

<div style="Background-color:teal; width:100%; height:5%; float:left; padding:10px">

<font size=5 color=yellow> <b> ADD NEW PRODUCT <b> </font>

</div>

<table cellspacing=10>

<tr>

<td> Company Name </td>

<td> <input type=text name=cname size=20> </td>

</tr>

<tr>

<td> Model </td>

<td> <input type=text name=mdl size=20> </td>

</tr>

<tr>

<td> Color </td>

<td> <input type=text name=mclr size=20> </td>

</tr>

<tr>

<td> Price </td>

<td> <input type=text name=mprice size=20> </td>

</tr>

<tr>

<td> Image File </td>

<td> <input type=text name=mimage size=20> </td>

</tr>

<tr>

<td> Video File </td>

<td colspan=3> <input type=text name=mvideo size=20> </td>

</tr>

<tr>

<td> Quantity </td>

<td> <input type=text name=mqnt size=20> </td>

</tr>

<tr>

<td> Storage </td>

<td> <input type=text name=mstor size=50> </td>

</tr>

<tr>

<td> Camera </td>

<td> <input type=text name=mcam size=50> </td>

</tr>

<tr>

<td> Battery </td>

<td> <input type=text name=mbat size=50> </td>

</tr>

<tr>

<td> Display Size </td>

<td> <input type=text name=mdisp size=50> </td>

</tr>

<tr>

<td> Operating System </td>

<td> <input type=text name=mos size=50> </td>

</tr>

<tr>

<td> Weight </td>

<td> <input type=text name=mwght size=50> </td>

</tr>

<tr>

<td> Launch Date </td>

<td> <input type=text name=mdate size=50> </td>

</tr>

<tr>

<td> Technology </td>

<td> <input type=text name=mtech size=50> </td>

</tr>

<tr>

<td> SIM </td>

<td> <input type=text name=msim size=50> </td>

</tr>

<tr>

<td> Other Features </td>

<td> <input type=text name=mfeature size=50> </td>

</tr>

</table>

<hr color=darkblue size=2>

<input type=submit name=b1 value=SAVE style="background-color:crimson; color:white; width:8%; height:25px">

<input type=reset name=b2 value=RESET style="background-color:crimson; color:white; width:8%; height:25px">

<%

String r=request.getParameter("b1");

if("SAVE".equals(r))

{

int pc, quan;

String cpnm, modl, clr, img, vid, store, cam, batry, size, os, wght, ldate, tech, sim, ofeatures;

cpnm=request.getParameter("cname");

modl=request.getParameter("mdl");

clr=request.getParameter("mclr");

pc=Integer.parseInt(request.getParameter("mprice"));

img=request.getParameter("mimage");

vid=request.getParameter("mvideo");

quan=Integer.parseInt(request.getParameter("mqnt"));

store=request.getParameter("mstor");

cam=request.getParameter("mcam");

batry=request.getParameter("mbat");

size=request.getParameter("mdisp");

os=request.getParameter("mos");

wght=request.getParameter("mwght");

ldate=request.getParameter("mdate");

tech=request.getParameter("mtech");

sim=request.getParameter("msim");

ofeatures=request.getParameter("mfeature");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q1="insert into product values('"+cpnm+"','"+modl+"','"+clr+"',"+pc+",'"+img+"','"+vid+"',"+quan+",'"+store+"','"+cam+"','"+batry+"','"+size+"','"+os+"','"+wght+"','"+ldate+"','"+tech+"','"+sim+"','"+ofeatures+"')";

state.executeUpdate(q1);

out.write("<h2> Successfully Added </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</form>

</body>

</html>

**Product List Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue text=darkblue>

<center>

<h2 style="color:red">

<u> PRODUCT LIST </u>

</h2>

<table width=1000 border=2 cellpadding=10>

<tr>

<td> <h2> Company Name </h2> </td>

<td> <h2> Model </h2> </td>

<td> <h2> Colour </h2> </td>

<td> <h2> Price </h2> </td>

<td> <h2> Image file </h2> </td>

<td> <h2> Video File </h2> </td>

<td> <h2> Quantity </h2> </td>

<td> <h2> Storage </h2> </td>

<td> <h2> Camera </h2> </td>

<td> <h2> Battery </h2> </td>

<td> <h2> Display Size </h2> </td>

<td> <h2> Operating System </h2> </td>

<td> <h2> Weight </h2> </td>

<td> <h2> Launch date </h2> </td>

<td> <h2> Technology </h2> </td>

<td> <h2> Sim </h2> </td>

<td> <h2> Other Features </h2> </td>

</tr>

<%

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q= "select \* from product";

ResultSet res= state.executeQuery(q);

while(res.next())

{

%>

<tr>

<td> <% out.write(res.getString("company")); %> </td>

<td> <% out.write(res.getString("Model")); %> </td>

<td> <% out.write(res.getString("Color")); %> </td>

<td> <% out.write(res.getString("Price")); %> </td>

<td> <% out.write(res.getString("Image")); %> </td>

<td> <% out.write(res.getString("Video")); %> </td>

<td> <% out.write(res.getString("Quantity")); %> </td>

<td> <% out.write(res.getString("storage")); %> </td>

<td> <% out.write(res.getString("camera")); %> </td>

<td> <% out.write(res.getString("battery")); %> </td>

<td> <% out.write(res.getString("display\_size")); %> </td>

<td> <% out.write(res.getString("os")); %> </td>

<td> <% out.write(res.getString("weight")); %> </td>

<td> <% out.write(res.getString("launch\_date")); %> </td>

<td> <% out.write(res.getString("technology")); %> </td>

<td> <% out.write(res.getString("sim")); %> </td>

<td> <% out.write(res.getString("other\_features")); %> </td>

</tr>

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

%>

</table>

</center>

</body>

</html>

**Edit/ Delete Product Page**

<%@ page import="java.sql.\*" %>

<html>

<body bgcolor=powderblue>

<form>

<h2> <font color=darkblue> EDIT/ DELETE PRODUCT </font> </h2>

<hr>

<font color=blue size=4>

<table cellspacing=10>

<tr>

<td> Enter Company </td>

<td> <input type=text name=t1 size=20> </td>

</tr>

<tr>

<td> Enter Model </td>

<td> <input type=text name=t2 size=20> </td>

</tr>

<tr>

<td> Enter Color </td>

<td> <input type=text name=t3 size=20> </td>

</tr>

</table>

<input type=submit name=b1 value=SEARCH style="width:150;height:30;background-color:maroon; color:white; border:none"> </tr>

<%

String r1=request.getParameter("b1");

if("SEARCH".equals(r1))

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String co=request.getParameter("t1");

String md=request.getParameter("t2");

String cl=request.getParameter("t3");

String q1="select \* from product where company='"+co+"' and model='"+md+"' and color='"+cl+"'";

ResultSet res=state.executeQuery(q1);

if(!res.next())

out.write("Not Found");

else

{

%>

<hr>

<table cellspacing=20>

<tr>

<td> Company Name </td>

<td> <input type=text disabled name=t4 size=20 value="<%out.write(res.getString("Company"));%>"> </td>

</tr>

<tr>

<td> Model </td>

<td> <input type=text disabled name=t5 size=20 value="<% out.write(res.getString("model")); %>"> </td>

</tr>

<tr>

<td> Color </td>

<td> <input type=text disabled name=t6 size=20 value="<% out.write(res.getString("color")); %>"> </td>

</tr>

<tr>

<td> Price </td>

<td colspan=3> <input type=text name=t7 size=20 value="<% out.write(res.getString("price")); %>"> </td>

</tr>

<tr>

<td> Image File </td>

<td> <input type=text name=t8 size=50 value="<% out.write(res.getString("image")); %>"> </td>

</tr>

<tr>

<td> Video File </td>

<td> <input type=text name=t9 size=50 value="<% out.write(res.getString("video")); %>"> </td>

</tr>

<tr>

<td> Quantity </td>

<td> <input type=text name=t10 size=20 value="<% out.write(res.getString("quantity")); %>"> </td>

</tr>

<tr>

<td> Storage </td>

<td> <input type=text name=t11 size=50 value="<% out.write(res.getString("storage")); %>"> </td>

</tr>

<tr>

<td> Camera </td>

<td> <input type=text name=t12 size=50 value="<% out.write(res.getString("camera")); %>"> </td>

</tr>

<tr>

<td> Battery </td>

<td> <input type=text name=t13 size=50 value="<% out.write(res.getString("battery")); %>"> </td>

</tr>

<tr>

<td> Display Size </td>

<td> <input type=text name=t14 size=50 value="<% out.write(res.getString("display\_size")); %>"> </td>

</tr>

<tr>

<td> Operating System </td>

<td> <input type=text name=t15 size=50 value="<% out.write(res.getString("os")); %>"> </td>

</tr>

<tr>

<td> Weight </td>

<td> <input type=text name=t16 size=50 value="<% out.write(res.getString("weight")); %>"> </td>

</tr>

<tr>

<td> Launch Date </td>

<td> <input type=text name=t17 size=50 value="<% out.write(res.getString("launch\_date")); %>"> </td>

</tr>

<tr>

<td> Technology </td>

<td> <input type=text name=t18 size=50 value="<% out.write(res.getString("technology")); %>"> </td>

</tr>

<tr>

<td> SIM </td>

<td> <input type=text name=t19 size=50 value="<% out.write(res.getString("sim")); %>"> </td>

</tr>

<tr>

<td> Other Features </td>

<td> <input type=text name=t20 size=50 value="<% out.write(res.getString("other\_features")); %>"> </td>

</tr>

</table>

<hr>

<br>

<input type=submit name=b2 value=DELETE style="width:150;height:30;background-color:maroon; color:white; border:none">

<input type=submit name=b3 value=UPDATE style="width:150;height:30;background-color:maroon; color:white; border:none">

<%

}

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

String r2=request.getParameter("b2");

if("DELETE".equals(r2))

{

String co2=request.getParameter("t4");

String md2=request.getParameter("t5");

String cl2=request.getParameter("t6");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q2="delete from product where company='"+co2+"' and model='"+md2+"' and color='"+cl2+"'";

state.executeUpdate(q2);

out.write("<h2> Successfully Deleted </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

String r3=request.getParameter("b3");

if("UPDATE".equals(r3))

{

String co3=request.getParameter("t4");

String md3=request.getParameter("t5");

String cl3=request.getParameter("t6");

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("Jdbc:Oracle:thin:@localhost:1521:xe", "system","shiv789");

Statement state=con.createStatement();

String q3="update product set price='"+request.getParameter("t7")+"',image='"+request.getParameter("t8")+"',video='"+request.getParameter("t9")+"',quantity='"+request.getParameter("t10")+"',storage='"+request.getParameter("t11")+"',camera='"+request.getParameter("t12")+"',battery='"+request.getParameter("t13")+"',display\_size='"+request.getParameter("t14")+"',os='"+request.getParameter("t15")+"',weight='"+request.getParameter("t16")+"',launch\_date='"+request.getParameter("t17")+"',technology='"+request.getParameter("t18")+"',sim='"+request.getParameter("t19")+"',other\_features='"+request.getParameter("t20")+"' where company='"+co3+"' and model='"+md3+"' and color='"+cl3+"'";

state.executeUpdate(q3);

out.write("<h2> Successfully Updated </h2>");

}

catch(Exception obj)

{

out.write(obj.getMessage());

}

}

%>

</form>

</body>

</html>

**Comparison Page**

<%@page import="java.sql.\*"%>

<html>

<body bgcolor=skyblue text=RED style="margin:0">

<form>

<div style="Background-color:teal;width:100%;height:5%; padding:15px">

<font color=yellow size=5><B>

<center>

COMPARISON</B></font></div>

</center>

<div style="Background-color:white;width:48.8%;height:87%; float:left;margin-right:5px;margin-left:6px;padding:5px;margin-top:5px">

<iframe name=frmp1 width=100% height=100% src=prod1.jsp>

</iframe>

</div>

<div style="Background-color:white;width:48.8%;height:87%; float:left;padding:5px;margin-top:5px">

<iframe name=frmp2 width=100% height=100% src=prod2.jsp>

</iframe>

</div>

</center>

</div>

Prod1.jsp

<%@page import="java.sql.\*"%>

<html>

<body style="margin:0" bgcolor=white text=darkblue>

<form>

<br>

<center><B><font color=crimson>PRODUCT 1</font></B></center>

<p>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product";

ResultSet res=state.executeQuery(q);

%>

<table align=center>

<tr>

<td>Company</td>

<td><select name=pt11 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("company"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</select>

</td></tr>

<tr>

<td>Model</td>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product";

ResultSet res=state.executeQuery(q);

%>

<td><select name=pt12 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("model"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</td>

</tr>

<tr>

<td>Color</td>

<td>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select distinct(color) from product";

ResultSet res=state.executeQuery(q);

%>

<select name=pt13 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("color"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</td>

</tr>

</table>

<p>

<center>

<input type=submit value="SHOW DETAIL" name=b1 size=20>

<hr>

<%

String r=request.getParameter("b1");

if("SHOW DETAIL".equals(r))

{

String x=request.getParameter("pt11");

String y=request.getParameter("pt12");

String z=request.getParameter("pt13");

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product where company='"+x+"' and model='"+y+"' and color='"+z+"'" ;

ResultSet res=state.executeQuery(q);

if(!res.next())

{

out.write("<h2>NOT FOUND</h2>");

}

else

{

%>

<center>

<table cellspacing=7>

<tr><td>Features</td></tr>

<tr><td><input type=text name=pt14 value="<%out.write(res.getString("features"));%>" size=80></td></tr>

<tr><td>Price</td></tr>

<tr><td><input type=text name=pt15 value="<%out.write(res.getString("price"));%>" size=50></td></tr>

</table>

</center>

<br>

<%

session.setAttribute("fname1",res.getString("video"));

%>

<img src=<%out.write(res.getString("image"));%> width=35% height=40% style="margin-left:20px;float:left">

<div style="background-color:lightgrey;width:45%;height:40%;margin-left:20px;float:left">

<iframe name=frm3 width=100% height=100%>

</iframe>

</div>

<br><br><br><br><br><br>

<a href=video1.jsp target=frm3><font color=blue>PLAY VIDEO</font></a>

<%

}

}

catch(Exception e)

{

out.write(e.getMessage());

}

}

%>

</center>

</div>

</form>

</body>

</html>

Prod2.jsp

<%@page import="java.sql.\*"%>

<html>

<body style="margin:0" bgcolor=white text=darkblue>

<form>

<br>

<center><B><font color=crimson>PRODUCT 1</font></B></center>

<p>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product";

ResultSet res=state.executeQuery(q);

%>

<table align=center>

<tr>

<td>Company</td>

<td><select name=pt11 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("company"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</select>

</td></tr>

<tr>

<td>Model</td>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product";

ResultSet res=state.executeQuery(q);

%>

<td><select name=pt12 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("model"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</td>

</tr>

<tr>

<td>Color</td>

<td>

<%

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select distinct(color) from product";

ResultSet res=state.executeQuery(q);

%>

<select name=pt13 style="width:250">

<%

while(res.next())

{

%>

<option><%out.write(res.getString("color"));%></option>

<%

}

}

catch(Exception ob)

{

out.write(ob.getMessage());

}

%>

</td>

</tr>

</table>

<p>

<center>

<input type=submit value="SHOW DETAIL" name=b1 size=20>

<hr>

<%

String r=request.getParameter("b1");

if("SHOW DETAIL".equals(r))

{

String x=request.getParameter("pt11");

String y=request.getParameter("pt12");

String z=request.getParameter("pt13");

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:mob");

Statement state=con.createStatement();

String q="Select \* from product where company='"+x+"' and model='"+y+"' and color='"+z+"'" ;

ResultSet res=state.executeQuery(q);

if(!res.next())

{

out.write("<h2>NOT FOUND</h2>");

}

else

{

%>

<center>

<table cellspacing=7>

<tr><td>Features</td></tr>

<tr><td><input type=text name=pt14 value="<%out.write(res.getString("features"));%>" size=80></td></tr>

<tr><td>Price</td></tr>

<tr><td><input type=text name=pt15 value="<%out.write(res.getString("price"));%>" size=50></td></tr>

</table>

</center>

<br>

<%

session.setAttribute("fname1",res.getString("video"));

%>

<img src=<%out.write(res.getString("image"));%> width=35% height=40% style="margin-left:20px;float:left">

<div style="background-color:lightgrey;width:45%;height:40%;margin-left:20px;float:left">

<iframe name=frm3 width=100% height=100%>

</iframe>

</div>

<br><br><br><br><br><br>

<a href=video1.jsp target=frm3><font color=blue>PLAY VIDEO</font></a>

<%

}

}

catch(Exception e)

{

out.write(e.getMessage());

}

}

%>

</center>

</div>

</form>

</body>

</html>

**CHAPTER 4**

**INPUT AND OUTPUT SCREENS**

**Home Page**



**Fig 4.1**

**Registration Form**



**Fig 4.2**

**Change Profile form**

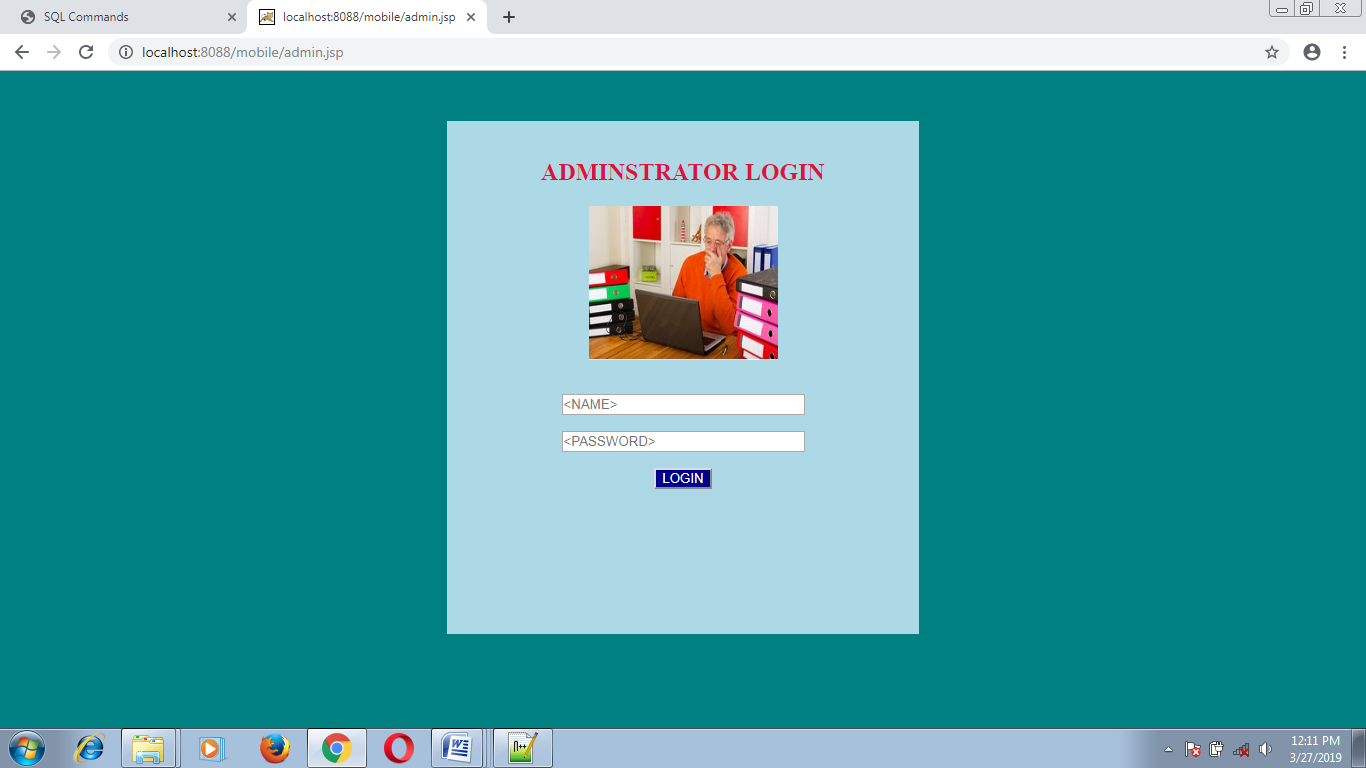


**Fig 4.3**



Fig 4.4

**Admin Login Form**



**Fig 4.5**

**Authority Page**



Fig 4.6

**Delete Registration Form**



Fig 4.7



**Registration List**



**Fig 4.8**

**Add New Product Form**



**Fig 4.9**

**Edit/Delete Product Form**



**Fig 4.10**



**Product List Page**



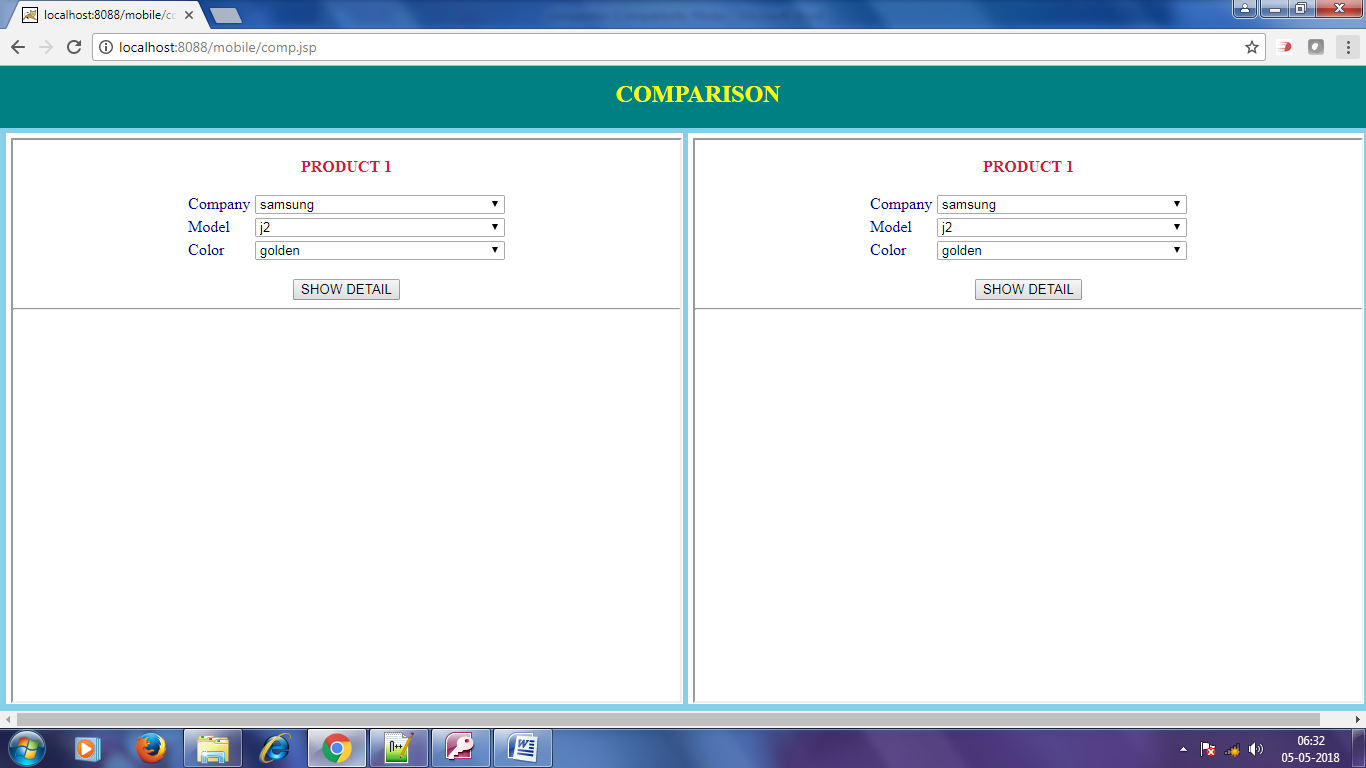
**Fig 4.11**

**Change Password Form**



Fig 4.11

**Comparison Page**



## Fig 4.12

## References

1. The Trost and A. Žemva, "Configurable hardware components generator in Javascript," *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, 2015, pp. 96-99, doi: 10.1109/MECO.2015.7181876.
2. A. Kumar and S. P. Panda, "A Survey: How Javascript Pitches in IT-World," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 248-251, doi: 10.1109/COMITCon.2019.8862251
3. Y. Liu, "JSOptimizer: An Extensible Framework for JavaScript Program Optimization," *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, pp. 168-170, doi: 10.1109/ICSE-Companion.2019.00069.
4. H. Park, W. Jung and S. Moon, "Javascript ahead-of-time compilation for embedded web platform," *2015 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia)*, 2015, pp. 1-9, doi: 10.1109/ESTIMedia.2015.7351768.
5. S. Delcev and D. Draskovic, "Modern JavaScript frameworks: A Survey Study," *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, 2018, pp. 106-109, doi: 10.1109/ZINC.2018.8448444.
6. M. Akbar, F. N. Azizah and G. A. P. Saptawati, "Integration of HTML tables in web pages," *2015 International Conference on Data and Software Engineering (ICoDSE)*, 2015, pp. 132-137, doi: 10.1109/ICODSE.2015.7436985.
7. S. -. Lim and Y. -. Ng, "Extracting structures of HTML documents," *Proceedings Twelfth International Conference on Information Networking (ICOIN-12)*, 1998, pp. 420-426, doi: 10.1109/ICOIN.1998.648420
8. Y. C. Chou and H. C. Liao, "A Webpage Data Hiding Method by Using Tag and CSS Attribute Setting," *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2014, pp. 122-125, doi: 10.1109/IIH-MSP.2014.37.
9. Vamsi Krishna Myalapalli and Bhupati Lohith Ravi Teja, "High performance PL/SQL programming," *2015 International Conference on Pervasive Computing (ICPC)*, 2015, pp. 1-5, doi: 10.1109/PERVASIVE.2015.7087001.
10. T. Q. Dam, S. Cheon and Y. Won, "On the IO Characteristics of the SQLite Transactions," *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2016, pp. 214-224, doi: 10.1109/MobileSoft.2016.047.
11. J. Chou, L. Chen, H. Ding, J. Tu and B. Xu, "A Method of Optimizing React JS Based on Greedy Strategy," *2013 10th Web Information System and Application Conference*, 2013, pp. 176-179, doi: 10.1109/WISA.2013.41.
12. T. Naumovic, M. Despotovic-Zrakic, B. Radenkovic, L. Zivojinovic and I. Jezdovic, "Development of a Continuous System Simulation Engine in Javascript Programing Language," *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2020, pp. 1-5, doi: 10.1109/INFOTEH48170.2020.9066334.
13. F. Rigueira, J. Bernardino and I. Pedrosa, "Extraction of information from log files Using Javascript Programming and Tableau," *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1-7, doi: 10.23919/CISTI49556.2020.9140844.
14. M. A. P. Subali and S. Rochimah, "A new model for measuring the complexity of SQL commands," *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 1-5, doi: 10.1109/ICITEED.2018.8534782.
15. Y. Guo, N. Li, J. Offutt and A. Motro, "Automatically Repairing SQL Faults," *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2018, pp. 500-511, doi: 10.1109/QRS.2018.00063.