

SportyShoes

Source code with output Screenshots

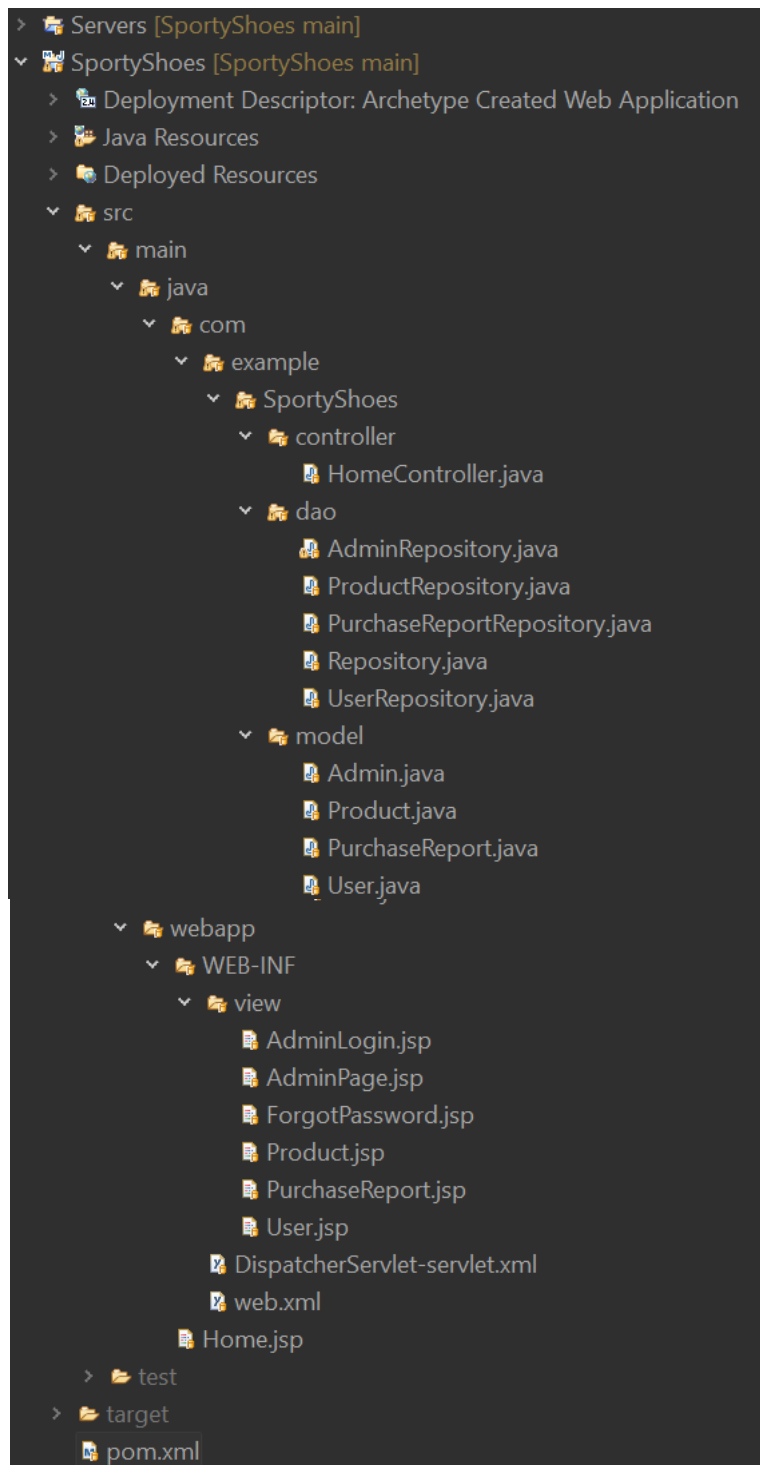
Project submitted by:

Ramakrishna G. Kannari

Email: rkannari@teksystems.com

Phone: +918296242431

- Project Structure and Packages



pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>SportyShoes</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>SportyShoes Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->

<dependency>
```

```
<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.27.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.3.5</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.23</version>

</dependency>

<dependency>
```

```
<groupId>org.glassfish.web</groupId>

<artifactId>javax.servlet.jsp.jstl</artifactId>

<version>1.2.1</version>

</dependency>

<dependency>

<groupId>javax.servlet.jsp.jstl</groupId>

<artifactId>javax.servlet.jsp.jstl-api</artifactId>

<version>1.2.1</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>SportyShoes</finalName>

<pluginManagement>

<!-- lock down plugins versions to avoid using Maven defaults (may be moved
to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_war_packaging -->

<plugin>
```

```
<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>
```

DispatcherServlet-servlet.xml

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

<context:component-scan
base-package="com.example.SportyShoes.controller" />

<context:component-scan
base-package="com.example.SportyShoes.repository" />

<context:component-scan
base-package="com.example.SportyShoes.dao" />

<bean name="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix">
<value>/WEB-INF/view/</value>
</property>

<property name="suffix">
<value>.jsp</value>
</property>
</bean>

<bean name="datasource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource"
p:driverClassName="com.mysql.jdbc.Driver"
```

```
p:url="jdbc:mysql://localhost:3306/sporty_shoes" p:username="root"
p:password="root" />

<bean name="sessionfactory"
class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
<property name="dataSource" ref="datasource" />
<property name="hibernateProperties">
<props>
<prop key="hibernate.dialect">org.hibernate.dialect.MySQL57Dialect</prop>
<prop key="hibernate.show_sql">true</prop>
</props>
</property>
<property name="annotatedClasses">
<list>
<value>com.example.SportyShoes.model.Admin</value>
<value>com.example.SportyShoes.model.User</value>
<value>com.example.SportyShoes.model.Product</value>
<value>com.example.SportyShoes.model.PurchaseReport</value>
</list>
</property>
</bean>

<bean name="hibernateTemplate"
class="org.springframework.orm.hibernate5.HibernateTemplate"
p:sessionFactory-ref="sessionfactory" />

<bean name="transactionManager"
class="org.springframework.orm.hibernate5.HibernateTransactionManager"
p:sessionFactory-ref="sessionfactory" />
<tx:annotation-driven />
</beans>
```


Web.xml

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:web="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd
http://xmlns.jcp.org/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" id="WebApp_ID" version="5.0">

<display-name>Archetype Created Web Application</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>Home.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>DispatcherServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>DispatcherServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>
```

model.Admin.java

```
package com.example.SportyShoes.model;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "admin")
public class Admin implements Serializable{

    private static final long serialVersionUID = 1L;

    @Id
    private String email;
    private String password;

    public Admin() {
        super();
    }

    public Admin(String email, String password) {
        super();
        this.email = email;
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public String toString() {
        return String.format("Admin [email=%s, password=%s]",
            email, password);
    }
}
```

}

}

model.User.java

```
package com.example.SportyShoes.model;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="user")
public class User implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int userId;
    private String userName;
    private String userEmail;
    private int userAge;
    private String userGender;
    private String userAddress;

    public User() {
        super();
    }

    public User(String userName, String userEmail, int userAge, String
userGender, String userAddress) {
        super();
        this.userName = userName;
        this.userEmail = userEmail;
        this.userAge = userAge;
        this.userGender = userGender;
        this.userAddress = userAddress;
    }

    public int getUserId() {
        return userId;
    }

    public void setUserId(int userId) {
        this.userId = userId;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
```

```

        this.userName = userName;
    }

    public String getUserEmail() {
        return userEmail;
    }

    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }

    public int getUserAge() {
        return userAge;
    }

    public void setUserAge(int userAge) {
        this.userAge = userAge;
    }

    public String getUserGender() {
        return userGender;
    }

    public void setUserGender(String userGender) {
        this.userGender = userGender;
    }

    public String getUserAddress() {
        return userAddress;
    }

    public void setUserAddress(String userAddress) {
        this.userAddress = userAddress;
    }

    @Override
    public String toString() {
        return String.format("User [userId=%s, userName=%s, userEmail=%s, userAge=%s, userGender=%s, userAddress=%s]",
            userId, userName, userEmail, userAge,
            userGender, userAddress);
    }
}

```

model.Product.java

```
package com.example.SportyShoes.model;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="product")
public class Product implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int productId;
    private String productName;
    private int productPrice;
    private String productDiscription;

    public Product() {
        super();
    }

    public Product(String productName, int productPrice, String
productDiscription) {
        super();
        this.productName = productName;
        this.productPrice = productPrice;
        this.productDiscription = productDiscription;
    }

    public int getProductId() {
        return productId;
    }

    public void setProductId(int productId) {
        this.productId = productId;
    }

    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public int getProductPrice() {
```

```
        return productPrice;
    }

    public void setProductPrice(int productPrice) {
        this.productPrice = productPrice;
    }

    public String getProductDiscription() {
        return productDiscription;
    }

    public void setProductDiscription(String productDiscription) {
        this.productDiscription = productDiscription;
    }

    @Override
    public String toString() {
        return String.format("Product [productId=%s, productName=%s, productprice=%s, productDiscription=%s]", productId, productName, productPrice, productDiscription);
    }
}
```

model.PurchaseReport.java

```
package com.example.SportyShoes.model;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="purchase_report")
public class PurchaseReport implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int reportId;
    private String reportProductName;
    private String reportUserEmail;
    private String reportDate;
    private int reportPrice;

    public PurchaseReport() {
        super();
    }

    public PurchaseReport(String reportProductName, String
reportUserEmail, String reportDate, int reportPrice) {
        super();
        this.reportProductName = reportProductName;
        this.reportUserEmail = reportUserEmail;
        this.reportDate = reportDate;
        this.reportPrice = reportPrice;
    }

    public int getReportId() {
        return reportId;
    }

    public void setReportId(int reportId) {
        this.reportId = reportId;
    }

    public String getReportProductName() {
        return reportProductName;
    }

    public void setReportProductName(String reportProductName) {
        this.reportProductName = reportProductName;
    }
}
```



```

    }

    public String getReportUserEmail() {
        return reportUserEmail;
    }

    public void setReportUserEmail(String reportUserEmail) {
        this.reportUserEmail = reportUserEmail;
    }

    public String getReportDate() {
        return reportDate;
    }

    public void setReportDate(String reportDate) {
        this.reportDate = reportDate;
    }

    public int getReportPrice() {
        return reportPrice;
    }

    public void setReportPrice(int reportPrice) {
        this.reportPrice = reportPrice;
    }

    public static long getSerialversionuid() {
        return serialVersionUID;
    }

    @Override
    public String toString() {
        return String.format(
            "PurchaseReport [reportId=%s,
reportProductName=%s, reportUserEmail=%s, reportDate=%s, reportPrice=%s]",
            reportId, reportProductName,
reportUserEmail, reportDate, reportPrice);
    }
}

```

dao.Repository.java

```
package com.example.SportyShoes.dao;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Component;

import com.example.SportyShoes.model.Admin;

@Component
@SuppressWarnings("deprecation")
public class Repository {

    @Autowired
    private HibernateTemplate hibernateTemplate;

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate
hibernateTemplate) {
        this.hibernateTemplate=hibernateTemplate;
    }

    public boolean forgotPassword(String email,String password) {

        Session session =
hibernateTemplate.getSessionFactory().openSession();
        Transaction tx = session.beginTransaction();
        @SuppressWarnings("unchecked")
        Query<Admin> q = session.createQuery("update Admin set
password=:p where email=:e");
        q.setParameter("e", email);
        q.setParameter("p",password);

        int status=q.executeUpdate();
        System.out.println(status);
        tx.commit();
        return status>0?true:false;

    }

}
```

dao.AdminRepository.java

```
package com.example.SportyShoes.dao;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Component;

import com.example.SportyShoes.model.Admin;

@Component
@SuppressWarnings("deprecation")
public class AdminRepository {

    @Autowired
    private HibernateTemplate hibernateTemplate;

    public HibernateTemplate gethibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate
hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }

    public boolean verifyAdmin(Admin a) {
        Session session =
hibernateTemplate.getSessionFactory().openSession();
        Transaction transaction = session.beginTransaction();

        String hql = "from Admin where email=:email and
password=:password";
        Query<Admin> query = session.createQuery(hql, Admin.class);
        query.setParameter("email", a.getEmail());
        query.setParameter("password", a.getPassword());
        List<Admin> results = query.getResultList();
        transaction.commit();

        return results.size()>0;
    }
}
```

dao.ProductRepository.java

```
package com.example.SportyShoes.dao;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Component;

import com.example.SportyShoes.model.Product;

@Component
public class ProductRepository {

    @Autowired
    HibernateTemplate hibernateTemplate;

    public ProductRepository() {
        super();
    }

    public ProductRepository(HibernateTemplate hibernateTemplate) {
        super();
        this.hibernateTemplate = hibernateTemplate;
    }

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate
hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }

    public List<Product> getAllProducts() {
        List<Product> productList =
hibernateTemplate.loadAll(Product.class);
        return productList;
    }

    public Boolean deleteProduct(int id) {
        try {
            Session session =
hibernateTemplate.getSessionFactory().openSession();
            Transaction transaction =
session.beginTransaction();
            Product p = session.get(Product.class, id);
            session.delete(p);
            transaction.commit();
            session.close();
        } catch (Exception e) {
            return false;
        }
    }
}
```

```
        return true;
    }
}
```

dao.UserRepository.java

```
package com.example.SportyShoes.dao;

import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Component;

import com.example.SportyShoes.model.User;

@Component
@SuppressWarnings("deprecation")
public class UserRepository {

    @Autowired
    HibernateTemplate hibernateTemplate;

    public UserRepository() {
        super();
    }

    public UserRepository(HibernateTemplate hibernateTemplate) {
        super();
        this.hibernateTemplate = hibernateTemplate;
    }

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate
hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }

    public List<User> getAllUsers() {
        List<User> userList =
hibernateTemplate.loadAll(User.class);
        return userList;
    }

    public List<User> searchUser(String email){
        Session session =
hibernateTemplate.getSessionFactory().openSession();
        Transaction transaction = session.beginTransaction();

        String hql = "from User where userEmail=:email";
        Query<User> query = session.createQuery(hql,User.class);
        query.setParameter("email",email);
        List<User> user = query.getResultList();
        transaction.commit();
        session.close();
    }
}
```

```
        return user;
    }
}
```

dao.PurchaseReportRepository.java

```
package com.example.SportyShoes.dao;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Component;

import com.example.SportyShoes.model.PurchaseReport;

@Component
public class PurchaseReportRepository {

    @Autowired
    HibernateTemplate hibernateTemplate;

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate
hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }

    @SuppressWarnings("deprecation")
    public List<PurchaseReport> getReport() {

        @SuppressWarnings("unchecked")
        List<PurchaseReport> report=(List<PurchaseReport>)
hibernateTemplate.find("from PurchaseReport order by
reportProductName,reportDate");

        return report;
    }
}
```


controller.HomeController.java

```
package com.example.SportyShoes.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;

import com.example.SportyShoes.dao.AdminRepository;
import com.example.SportyShoes.dao.ProductRepository;
import com.example.SportyShoes.dao.PurchaseReportRepository;
import com.example.SportyShoes.dao.Repository;
import com.example.SportyShoes.dao.UserRepository;
import com.example.SportyShoes.model.Admin;

@Controller
public class HomeController {

    @Autowired
    AdminRepository adminRepository;
    @Autowired
    UserRepository userRepository;
    @Autowired
    ProductRepository productRepository;
    @Autowired
    PurchaseReportRepository purchaseReportRepository;
    @Autowired
    Repository repository;

    @RequestMapping("/")
    public String showHome() {
        return "Home";
    }

    @RequestMapping(value="adminLogin",method = RequestMethod.GET)
    public String adminLoginPage() {
        return "AdminLogin";
    }

    @RequestMapping(value = "forgotPassword",method=RequestMethod.GET)
    public String forgotPassword() {
        return "ForgotPassword";
    }

    @RequestMapping(value="adminForgotPassword",method=RequestMethod.POST)
    public String adminForgotPasswordPage(@RequestParam("email")String email, @RequestParam("password")String password,ModelMap map) {
        if(repository.forgotPassword(email, password))
            map.addAttribute("message","Password Updated");
        else
            map.addAttribute("message","Invalid Details");
        return "ForgotPassword";
    }
}
```

```

    @RequestMapping(value="adminPage",method=RequestMethod.POST)
    public String adminPage(@RequestParam("email")String email,
    @RequestParam("password")String password,ModelMap map) {

        if(adminRepository.verifyAdmin(new Admin(email,password)))
        {

            return "AdminPage";
        }
        else
        {
            map.addAttribute("message","Invalid Details");
            return "AdminLogin";
        }
    }

    @RequestMapping(value="user",method=RequestMethod.GET)
    public String getAllUsers(ModelMap map) {
        map.addAttribute("useList",userRepository.getAllUsers());
        return "User";
    }

    @RequestMapping(value="searchUser",method=RequestMethod.GET)
    public String searchUser(@RequestParam("email")String
email,ModelMap map) {

        map.addAttribute("userList",userRepository.searchUser(email));
        return "User";
    }

    @RequestMapping(value="product",method=RequestMethod.GET)
    public String getAllProducts(ModelMap map) {

        map.addAttribute("productList",productRepository.getAllProducts());
        return "Product";
    }

    @RequestMapping(value="deleteProduct",method=RequestMethod.GET)
    public String deleteProduct(@RequestParam("id")int id,ModelMap map)
{

        if(productRepository.deleteProduct(id))
            map.addAttribute("message","Product Deleted
Successfully");
        else
            map.addAttribute("message","Try after few minutes");

        map.addAttribute("productList",productRepository.getAllProducts());

        return "Product";
    }
}

```

```
@RequestMapping(value="purchaseReport",method=RequestMethod.GET)
public String getReport(ModelMap map) {

    map.addAttribute("report",purchaseReportRepository.getReport());
    return "PurchaseReport";

}
```

Home.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<meta charset="utf-8">

</head>

<body >

<h1 align="center">E-Commerce Sporty Shoes</h1><br><br>
<h2 align="center"><span class="badge badge-success">Developed By @
Ramakrishna G. Kannari</span></h2>
<p align="center"><a href="adminLogin" style="font-size:25px;">Admin
Login</a></p>
</body>
</html>
```

AdminLogin.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
</head>
<body >
    <br>
    <a href=Home.jsp
        style="color: black; text-decoration: none; font-size:
35px; font-weight: bold;">Sporty
        Shoes</a>
    <br>
    <br>

    <h2>Admin Login</h2>

    <div>
        <form action="adminPage" method=post>
            <label for=email>Email :-</label>    <input
type="email" name=email id=email /><br>
            <label for=pass>Password :-</label> <input
type="password" name=password id=pass /><br>
            <br>                                <input
type=submit value=submit /> <input type=reset />
            </form>
        </div>
        <br>
        <a href=forgotPassword style="font-size: 25; color: red;">Forgot
Password</a>
        <p style="color: green;">${message}</p>

</body>
</html>
```

AdminPage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<style>
div {
    border: 10px solid black;
    color: blue;
    font-size: 20px;
    height: 100px;
    width: 450px;
}

a {
    padding-left: 50px;
}
</style>
</head>
<body style="background-color: rgb(255, 252, 211)">
    <a href=Home.jsp
        style="color: black; text-decoration: none; font-size:
35px; font-weight: bold;">Sporty
        Shoes</a>
    <br>
    <br>
    <div>
        <br> <br>
        <a href="product" target="iframe">Products</a>
        <a href="user" target="iframe">Users</a>
        <a href="purchaseReport" target="iframe">Purchase
Report</a>
    </div>
    <br>
    <br>
    <iframe src="product" name="iframe"
        style="height: 400px; width: 900px;"></iframe>
</body>
</html>
```

ForgotPassword.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
</head>
<body >
    <br>
    <a href=Home.jsp
        style="color: black; text-decoration: none; font-size:
35px; font-weight: bold;">Sporty
        Shoes</a>
    <br>
    <br>
    <h2 style="color: blue;">Enter details to reset password</h2>
    <div>
        <form action=adminForgotPassword method=post>
            <label for=email>Email :-</label> <input
type="email" name=email
                id=email /><br>
            <br> <label for=pass>New Password :-</label> <input
                type="password" name=password id=pass /><br>
            <br> <input type=submit value=submit /> <input
type=reset />
            </form>
        </div>
        <br>
        <br>
        <p style="color: green;">${message}</p>
    </body>
</html>
```

Product.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<style>
table, th, td {
    border: 1px solid black;
    text-align: center;
}
</style>

</head>
<body >
    <h2>All Available Products</h2>
    <br>
    <table style="width: 100%">
        <tr>
            <th>Product Name</th>
            <th>Product Category</th>
            <th>Product Price</th>
        </tr>

        <c:forEach var="product" items="{productList}">

            <tr>
                <td>${product.productName}</td>
                <td>${product.productDiscription}</td>
                <td>${product.productPrice}</td>
                <td><a
href="deleteProduct?id=${product.productId}">Delete</a></td>
            </tr>

        </c:forEach>

    </table>
    <br>
    <br>
    <p style="color: green;">${message}</p>

</body>
</html>
```


PurchaseReport.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<style>
table, th, td {
    border: 1px solid black;
    text-align: center;
}
</style>

</head>
<body>

    <h2>Purchase Report</h2>
    <br>
    <table style="width: 100%">
        <tr>
            <th>Product name</th>
            <th>User Email</th>
            <th>Date</th>
            <th>Price</th>
        </tr>

        <c:forEach var="report" items="${report}">

            <tr>
                <td>${report.reportProductName}</td>
                <td>${report.reportUserEmail}</td>
                <td>${report.reportDate}</td>
                <td>${report.reportPrice}</td>
            </tr>

        </c:forEach>

    </table>

</body>
</html>
```

User.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<style>
table, th, td {
    border: 1px solid black;
    text-align: center;
}
</style>
</head>
<body>
    <h2>Users Details</h2>
    <br>

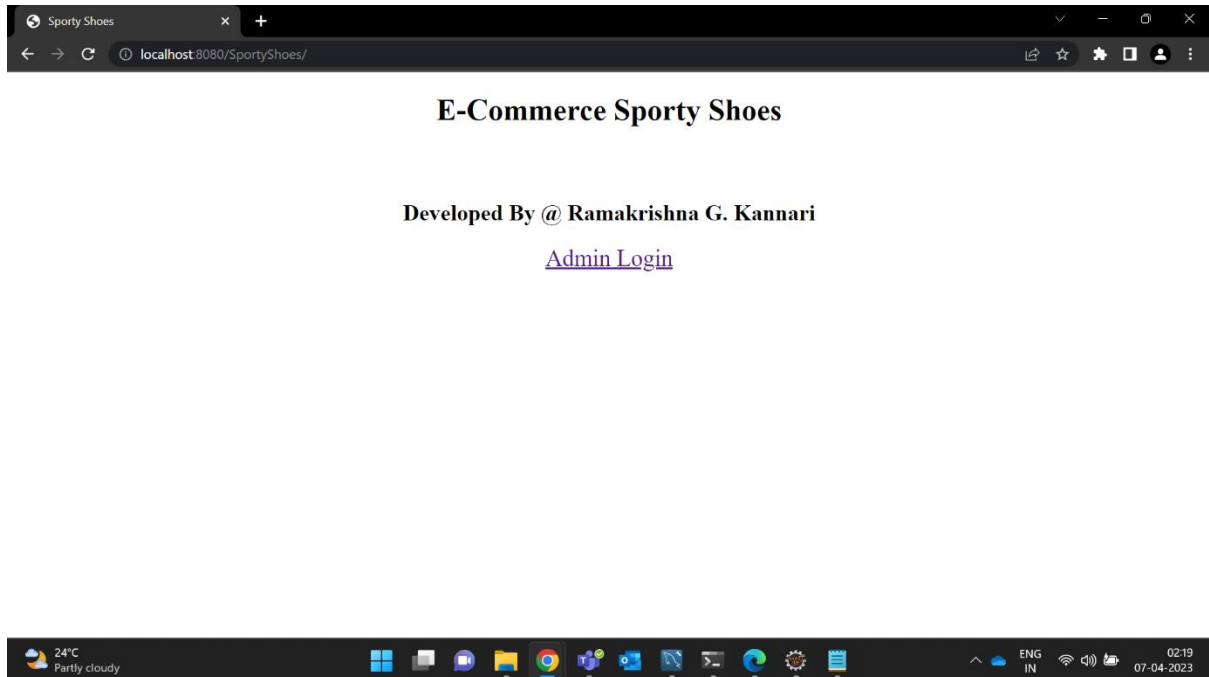
    <form action=searchUser>
        <label for="search">Search User By Email :-</label> <input
type="email" name="email" id="search" /> <input
type="submit"
        value="search" />
    </form>
    <br>
    <br>
    <table style="width: 100%">
        <tr>
            <th>Name</th>
            <th>Email</th>
            <th>Age</th>
            <th>Gender</th>
            <th>Address</th>
        </tr>

        <c:forEach var="user" items="${userList}">
            <tr>
                <td>${user.userName}</td>
                <td>${user.userEmail}</td>
                <td>${user.userAge}</td>
                <td>${user.userGender}</td>
                <td>${user.userAddress}</td>
            </tr>
        </c:forEach>
    </table>

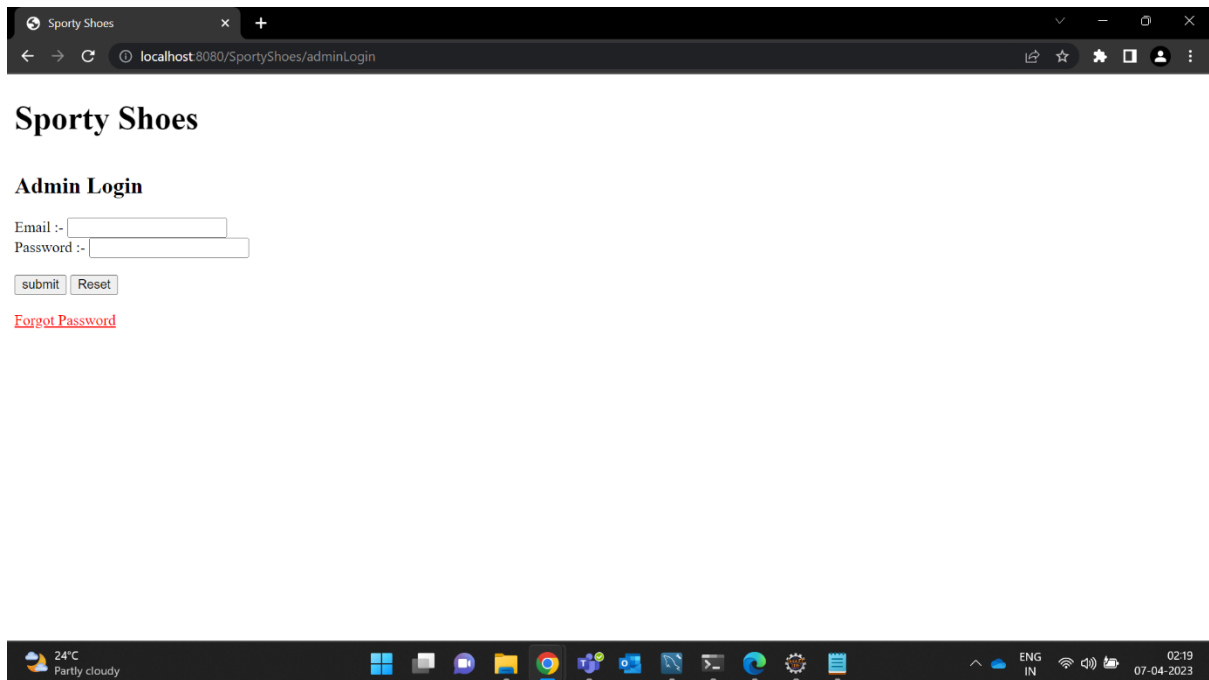
</body>
</html>
```

Output Screenshots:

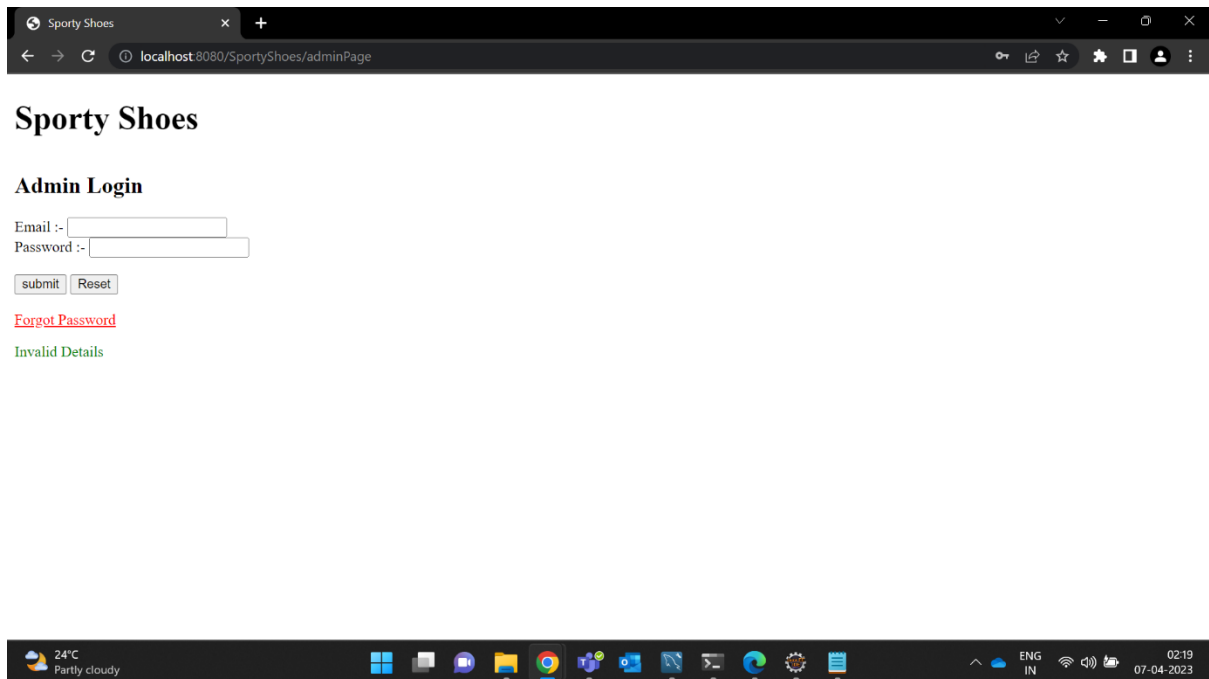
1. Home Page



2. Admin Login

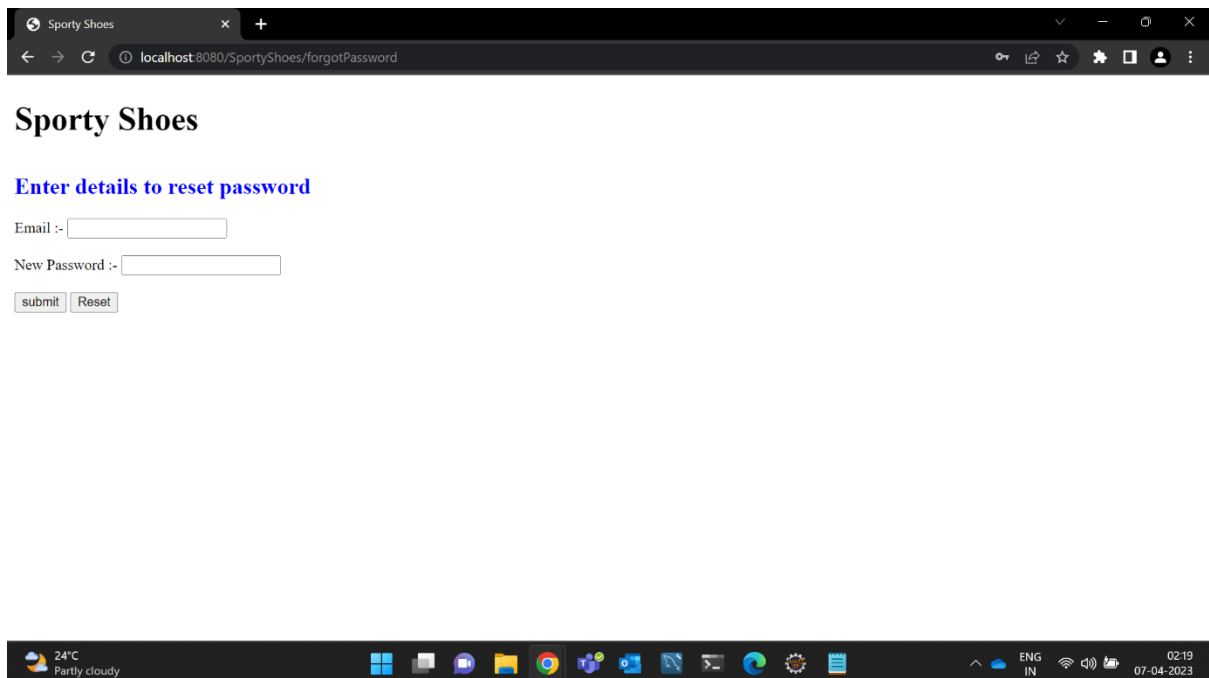


3. If Admin Provides Incorrect details



The screenshot shows a web browser window with the title 'Sporty Shoes' and the address bar displaying 'localhost:8080/SportyShoes/adminPage'. The page content includes the 'Sporty Shoes' logo, an 'Admin Login' section with input fields for 'Email :-' and 'Password :-', and buttons for 'submit' and 'Reset'. Below the login fields, there is a red link for 'Forgot Password' and a green message 'Invalid Details' indicating an authentication error. The Windows taskbar at the bottom shows the system clock as 02:19 on 07-04-2023.

4. Forgot Password Page



The screenshot shows a web browser window with the title 'Sporty Shoes' and the address bar displaying 'localhost:8080/SportyShoes/forgotPassword'. The page content includes the 'Sporty Shoes' logo, a blue heading 'Enter details to reset password', and input fields for 'Email :-' and 'New Password :-'. There are 'submit' and 'Reset' buttons at the bottom of the form. The Windows taskbar at the bottom shows the system clock as 02:19 on 07-04-2023.

5. Password Changed.

Sporty Shoes

[Enter details to reset password](#)

Email :-

New Password :-

Password Updated

6. Admin Page

Sporty Shoes

[Products](#) [Users](#) [Purchase Report](#)

All Available Products

Product Name	Product Category	Product Price	
Strickrer	This is men slipper for the normal walk	399	Delete
Bata	Slipper	499	Delete
Sparx	Snaeker	999	Delete
Bata	Sandal	699	Delete

7. Available Products Report

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/SportyShoes/adminPage'. The page title is 'Sporty Shoes'. Below the title, there is a navigation bar with three links: 'Products', 'Users', and 'Purchase Report'. The main content area is titled 'All Available Products' and contains a table with the following data:

Product Name	Product Category	Product Price	
Strickrer	This is men slipper for the normal walk	399	Delete
Bata	Slipper	499	Delete
Sparx	Snaeker	999	Delete
Bata	Sandal	699	Delete

The browser's status bar at the bottom shows the temperature as 24°C, the weather as 'Partly cloudy', and the date and time as 07-04-2023 02:20.

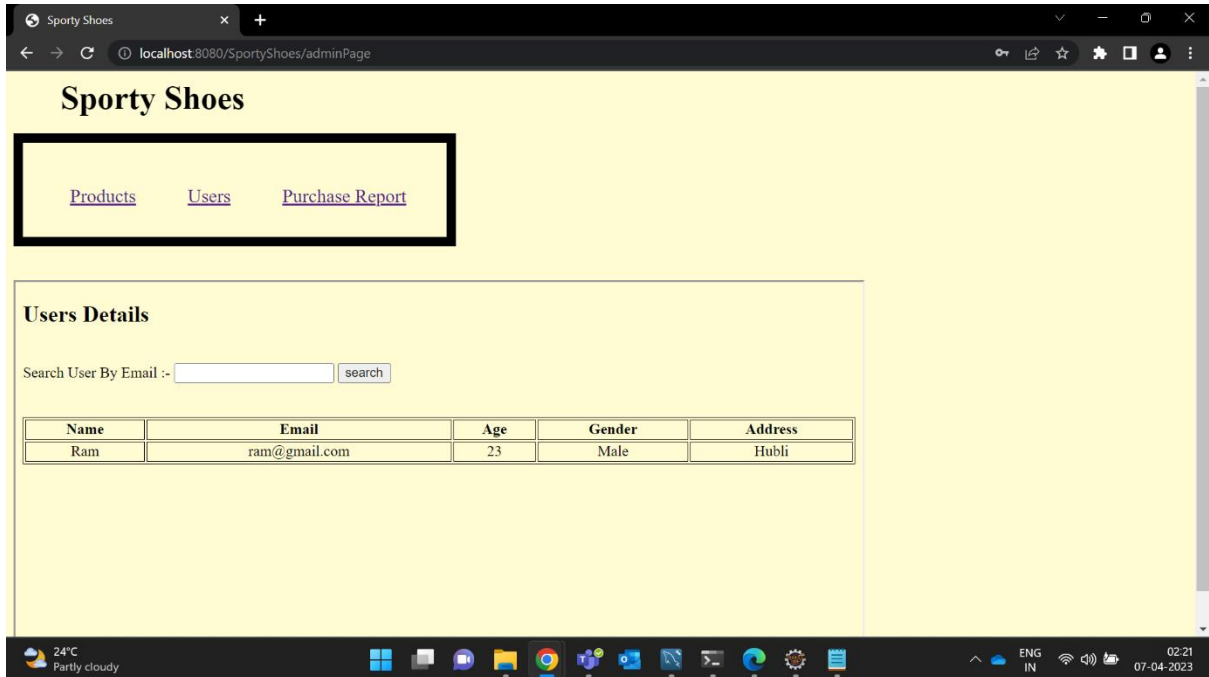
8. Delete Product

The screenshot shows the same web browser window as in the previous image, but with the address bar displaying 'localhost:8080/SportyShoes/deleteProduct?id=2'. The page title is 'Sporty Shoes'. Below the title, there is a navigation bar with three links: 'Products', 'Users', and 'Purchase Report'. The main content area is titled 'All Available Products' and contains a table with the following data:

Product Name	Product Category	Product Price	
Bata	Slipper	499	Delete
Sparx	Snaeker	999	Delete
Bata	Sandal	699	Delete

Below the table, there is a green message that says 'Product Deleted Successfully'. The browser's status bar at the bottom shows the temperature as 24°C, the weather as 'Partly cloudy', and the date and time as 07-04-2023 02:20.

9. Search user by email

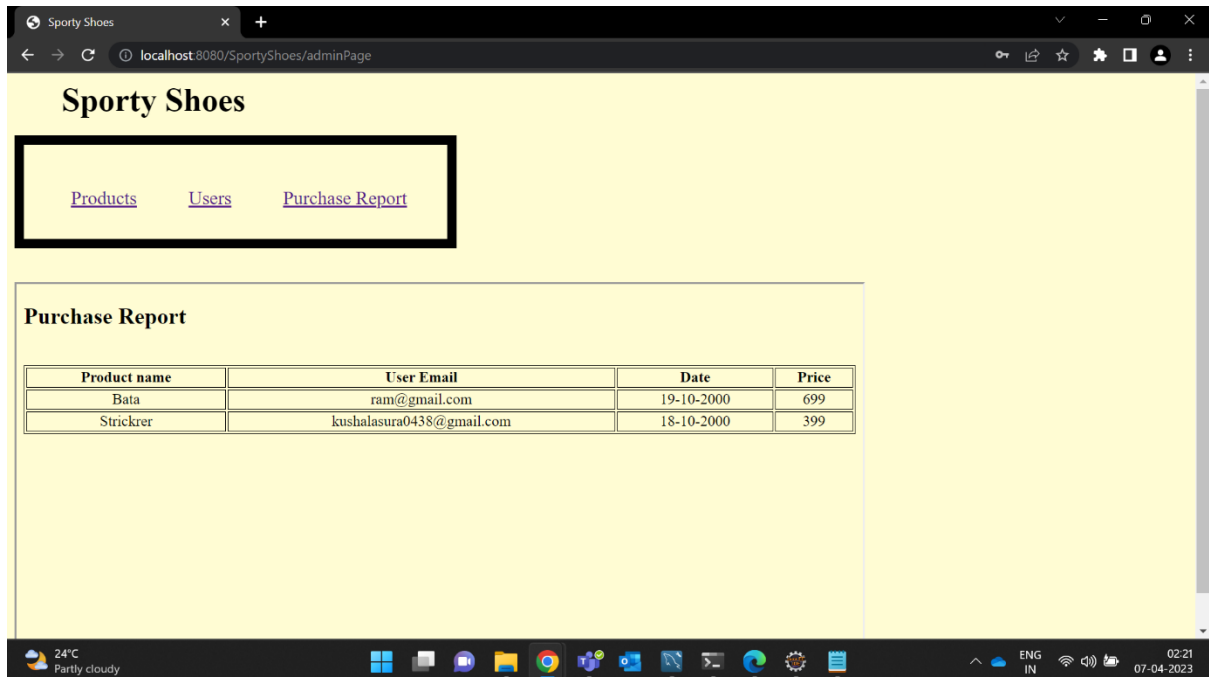


The screenshot shows a web browser window with the URL `localhost:8080/SportyShoes/adminPage`. The page has a yellow background and a black header bar. The header bar contains the text "Sporty Shoes" and a navigation menu with links for "Products", "Users", and "Purchase Report". The "Users" link is highlighted. Below the navigation menu, there is a section titled "Users Details". This section contains a search bar with the text "Search User By Email :-" and a "search" button. Below the search bar, there is a table with the following data:

Name	Email	Age	Gender	Address
Ram	ram@gmail.com	23	Male	Hubli

The Windows taskbar at the bottom shows the date and time as 07-04-2023 02:21.

10. Purchase Report



The screenshot shows the same web browser window as in the previous image, but the "Purchase Report" link in the navigation menu is highlighted. Below the navigation menu, there is a section titled "Purchase Report". This section contains a table with the following data:

Product name	User Email	Date	Price
Bata	ram@gmail.com	19-10-2000	699
Strickrer	kushalasure0438@gmail.com	18-10-2000	399

The Windows taskbar at the bottom shows the date and time as 07-04-2023 02:21.