

Controller

Main Controller

```
package projectkeep.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {
    @GetMapping("/")
    public String shoeHomePage() {
        System.out.println("main controller");
        return "index";
    }
}
```

UserController

```
package projectkeep.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
import projectkeep.exceptionHandler.UserNotFoundException;
import projectkeep.model.User;
import projectkeep.repository.UserRepository;
import projectkeep.service.UserService;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.List;

@Controller
public class UserController {
    @Autowired
    private UserService userService;

    @Autowired
    private UserRepository userRepository;

    //Start Crud Repository for Admin Management
    @GetMapping("/users")
    public String showUserList(Model model) {
```

```

        List<User> listUsers = userService.getList();
        model.addAttribute("listUsers", listUsers);
        return "users";
    }

    @GetMapping("/users/new")
    public String showNewForm(Model model) {
        model.addAttribute("user", new User());
        model.addAttribute("pageTitle", "Add New User");
        return "user_form";
    }

    @PostMapping("/users/save")
    public String saveUser(User user, RedirectAttributes
redirectAttributes) {
        userService.save(user);
        redirectAttributes.addFlashAttribute("message", "User Added
Successfully");
        return "redirect:/users";
    }

    @GetMapping("/users/edit/{id}")
    public String showEditForm(@PathVariable("id") Integer id, Model
model, RedirectAttributes redirectAttributes) {
        try {
            User user = userService.get(id);
            model.addAttribute("user", user);
            model.addAttribute("pageTitle", "Edit User(ID: " + id +
")");
            return "user_form";
        } catch (UserNotFoundException e) {
            redirectAttributes.addFlashAttribute("message", "Could not
find user with ID " + id);
            return "redirect:/users";
        }
    }

    @GetMapping("/users/delete/{id}")
    public String deleteUser(@PathVariable("id") Integer id,
RedirectAttributes redirectAttributes) {
        try {
            userService.delete(id);
            redirectAttributes.addFlashAttribute("message", "User with
ID " + id + " has been deleted");
        } catch (UserNotFoundException e) {
            redirectAttributes.addFlashAttribute("message", "Could not
find user with ID " + id);
        }
        return "redirect:/users";
    }
}

//End Crud Repository for User Management

```

```

//Controller for Lofin
@GetMapping("/login")
public String loginPage() {
    return "login_form";
}

@ResponseBody
@RequestMapping("/admin")
public ModelAndView userLogin(HttpServletRequest request,
    HttpServletResponse response, RedirectAttributes redirectAttributes) {
    ModelAndView modelAndView = new ModelAndView();
    String email = request.getParameter("email");
    if (userRepository.findLogin(email) != null) {
        modelAndView.setViewName("users");
    } else modelAndView.setViewName("login_form");
    return modelAndView;
}
}

```

Purchase Controller

```

package projectkeep.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import projectkeep.model.ShoeEntity;
import projectkeep.service.PurchaseService;

import java.util.List;

@Controller
public class PurchaseController {
    @Autowired
    private PurchaseService purchaseService;

    //Controller for Filter Category
    @RequestMapping("/report/search")
    public String viewResult(Model model, @Param("keyword") String
keyword) {

        List<ShoeEntity> listResults =
purchaseService.listAll(keyword);
        model.addAttribute("listShoes", listResults);
        model.addAttribute("keyword", keyword);
    }
}

```

```

        return "search_form";
    }
}

```

ShoeController

```

package projectkeep.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
import projectkeep.exceptionHandler.UserNotFoundException;
import projectkeep.model.ShoeEntity;
import projectkeep.repository.ShoeRepository;
import projectkeep.service.ShoeService;

import java.util.List;

@Controller
public class shoeController {
    @Autowired
    ShoeService shoeService;
    @Autowired
    ShoeRepository shoeRepository;

    @GetMapping("/report")
    public String getReport(Model model) {
        List<ShoeEntity> listShoes = shoeService.getShoesDetails();
        model.addAttribute("pageTitle", "Product Report");
        model.addAttribute("listShoes", listShoes);
        return "report";
    }

    @GetMapping("/shoeDetails")
    public String getShoeDetails(Model model) {
        List<ShoeEntity> listShoes = shoeService.getShoesDetails();
        model.addAttribute("listShoes", listShoes);
        return "product";
    }

    @GetMapping("/shoeDetails/new")
    public String showNewForm(Model model) {
        model.addAttribute("shoe", new ShoeEntity());
    }
}

```

```

        model.addAttribute("pageTitle", "Add New Product");
        return "add_shoes";
    }

    @PostMapping("/shoeDetails/save")
    public String saveUser(ShoeEntity shoeEntity, RedirectAttributes
redirectAttributes) {
        shoeService.save(shoeEntity);
        redirectAttributes.addFlashAttribute("message", "Product Saved
Successfully");
        return "redirect:/shoeDetails";
    }

    @GetMapping("/shoeDetails/edit/{id}")
    public String showEditForm(@PathVariable("id") Integer id, Model
model, RedirectAttributes redirectAttributes) {
        try {
            ShoeEntity shoeEntity = shoeService.get(id);
            model.addAttribute("shoe", shoeEntity);
            model.addAttribute("pageTitle", "Edit User(ID: " + id +
"");
            return "add_shoes";
        } catch (UserNotFoundException e) {
            redirectAttributes.addFlashAttribute("message", "Could not
find user with ID " + id);
            return "redirect:/shoeDetails";
        }
    }

    @GetMapping("/shoeDetails/delete/{id}")
    public String deleteUser(@PathVariable("id") Integer id,
RedirectAttributes redirectAttributes) {
        try {
            shoeService.delete(id);
            redirectAttributes.addFlashAttribute("message", "Product
with ID " + id + " has been deleted");
        } catch (UserNotFoundException e) {
            redirectAttributes.addFlashAttribute("message", "Could not
find Product with ID " + id);
        }
        return "redirect:/shoeDetails";
    }
    //End Crud Repository for User Management
}

```

Model

ShoeEntity

```
package projectkeep.model;

import javax.persistence.*;

@Entity
@Table(name = "shoe_details")
public class ShoeEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "shoe_id")
    private int id;
    @Column(name = "shoe_color")
    private String shoeColor;
    @Column(name = "shoe_size")
    private String shoeSize;
    @Column(name = "shoe_category")
    private String shoeCategory;
    @Column(nullable = true, name = "available_in_stock")
    private Integer availableInStock;
    @Column(name = "shoe_price")
    private int shoePrice;

    @Column(name = "date_added")
    private String dateAdded;
    @Transient
    private Integer remainingStocks;

    public ShoeEntity() {
    }

    public ShoeEntity(int id, String shoeColor, String shoeSize,
String shoeCategory, Integer availableInStock, int shoePrice, String
dateAdded, Integer remainingStocks) {
        this.id = id;
        this.shoeColor = shoeColor;
        this.shoeSize = shoeSize;
        this.shoeCategory = shoeCategory;
        this.availableInStock = availableInStock;
        this.shoePrice = shoePrice;
        this.dateAdded = dateAdded;
        this.remainingStocks = remainingStocks;
    }
}
```

```
    public ShoeEntity(String shoeColor, String shoeSize, String
shoeCategory, Integer availableInStock, int shoePrice, String
dateAdded, Integer remainingStocks) {
        this.shoeColor = shoeColor;
        this.shoeSize = shoeSize;
        this.shoeCategory = shoeCategory;
        this.availableInStock = availableInStock;
        this.shoePrice = shoePrice;
        this.dateAdded = dateAdded;
        this.remainingStocks = remainingStocks;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getShoeColor() {
        return shoeColor;
    }

    public void setShoeColor(String shoeColor) {
        this.shoeColor = shoeColor;
    }

    public String getShoeSize() {
        return shoeSize;
    }

    public void setShoeSize(String shoeSize) {
        this.shoeSize = shoeSize;
    }

    public String getShoeCategory() {
        return shoeCategory;
    }

    public void setShoeCategory(String shoeCategory) {
        this.shoeCategory = shoeCategory;
    }

    public Integer getAvailableInStock() {
        return availableInStock;
    }

    public void setAvailableInStock(Integer availableInStock) {
        this.availableInStock = availableInStock;
    }
}
```

```

public int getShoePrice() {
    return shoePrice;
}

public void setShoePrice(int shoePrice) {
    this.shoePrice = shoePrice;
}

public String getDateAdded() {
    return dateAdded;
}

public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
}

public Integer getRemainingStocks() {
    return remainingStocks;
}

public void setRemainingStocks(Integer remainingStocks) {
    this.remainingStocks = remainingStocks;
}

@Override
public String toString() {
    return "ShoeEntity{" +
        "id=" + id +
        ", shoeColor='" + shoeColor + '\'' +
        ", shoeSize='" + shoeSize + '\'' +
        ", shoeCategory='" + shoeCategory + '\'' +
        ", availableInStock=" + availableInStock +
        ", shoePrice=" + shoePrice +
        ", dateAdded='" + dateAdded + '\'' +
        ", remainingStocks=" + remainingStocks +
        '}';
}
}

```


User

```
package projectkeep.model;

import javax.persistence.*;

@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "email", nullable = false, unique = true, length =
45)
    private String email;
    @Column(length = 15, nullable = false)
    private String password;
    @Column(length = 45, nullable = false, name = "first_name")
    private String firstName;
    @Column(length = 45, nullable = false, name = "last_name")
    private String lastName;

    private boolean enabled;

    public User() {
    }

    public User(Integer id, String email, String password, String
firstName, String lastName) {
        this.id = id;
        this.email = email;
        this.password = password;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public User(String email, String password, String firstName,
String lastName) {
        this.email = email;
        this.password = password;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }
}
```

```

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;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public boolean isEnabled() {
    return enabled;
}

public void setEnabled(boolean enabled) {
    this.enabled = enabled;
}

@Override
public String toString() {
    return "User{" +
        "id=" + id +
        ", email='" + email + '\'' +
        ", password='" + password + '\'' +
        ", firstName='" + firstName + '\'' +
        ", lastName='" + lastName + '\'' +
        '}';
}
}

```

Repository

PurchaseRepository

```
package projectkeep.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import
org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import projectkeep.model.ShoeEntity;

import java.util.List;

@EnableJpaRepositories
public interface PurchaseRepository extends JpaRepository<ShoeEntity,
Integer> {

    @Query("SELECT s FROM ShoeEntity s WHERE s.shoeCategory LIKE
    %?1%")
    public List<ShoeEntity> search(String keyword);
}
```

ShoeRepository

```
package projectkeep.repository;

import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import projectkeep.model.ShoeEntity;

import java.util.List;

public interface ShoeRepository extends CrudRepository<ShoeEntity,
Integer> {
    public Long countById(Integer id);

    @Query("select shoe from ShoeEntity shoe where
    shoe.shoeCategory=?1")
    public List<ShoeEntity> searchByCategory(String shoeCategory);
}
```

UserRepository

```
package projectkeep.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;
import projectkeep.model.User;

@Repository
public interface UserRepository extends JpaRepository<User, Integer> {
    public Long countById(Integer id);

    @Query("select user from User user where user.email=?1")
    public User findLogin(String email);
}
```

Service

PurchaseService

```
package projectkeep.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import projectkeep.model.ShoeEntity;
import projectkeep.repository.PurchaseRepository;

import java.util.List;

@Service
public class PurchaseService {
    @Autowired
    private PurchaseRepository purchaseRepository;

    public List<ShoeEntity> listAll(String keyword) {
        if (keyword != null) {
            return purchaseRepository.search(keyword);
        }
        return purchaseRepository.findAll();
    }
}
```

UserService

```
package projectkeep.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import projectkeep.exceptionHandler.UserNotFoundException;
```

```

import projectkeep.model.User;
import projectkeep.repository.UserRepository;

import java.util.List;
import java.util.Optional;

@Service
public class UserService {
    @Autowired
    private UserRepository userRepository;

    public List<User> getList() {

        return (List<User>) userRepository.findAll();
    }

    public void save(User user) {
        userRepository.save(user);
    }

    public User get(Integer id) throws UserNotFoundException {
        Optional<User> result = userRepository.findById(id);
        if (result.isPresent()) {
            return result.get();
        }
        throw new UserNotFoundException("Could not find users with ID
" + id);
    }

    public void delete(Integer id) throws UserNotFoundException {
        Long count = userRepository.countById(id);
        if (count == null || count == 0) {
            throw new UserNotFoundException("Could not find users with
ID " + id);
        }
        userRepository.deleteById(id);
    }

    public User findLogin(String email) {
        return userRepository.findLogin(email);
    }
}

```

ShoeService

```
package projectkeep.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import projectkeep.exceptionHandler.UserNotFoundException;
import projectkeep.model.ShoeEntity;
import projectkeep.repository.ShoeRepository;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

@Service
public class ShoeService {
    @Autowired
    ShoeRepository shoeRepository;

    public List<ShoeEntity> getShoesDetails() {
        List<ShoeEntity> getShoes = new ArrayList<>();
        for (ShoeEntity shoeEntity : shoeRepository.findAll()) {
            getShoes.add(shoeEntity);
        }
        return getShoes;
    }

    public void save(ShoeEntity shoeEntity) {
        shoeRepository.save(shoeEntity);
    }

    public ShoeEntity get(Integer id) throws UserNotFoundException {
        Optional<ShoeEntity> result = shoeRepository.findById(id);
        if (result.isPresent()) {
            return result.get();
        }
        throw new UserNotFoundException("Could not find Product with
ID " + id);
    }

    public void delete(Integer id) throws UserNotFoundException {
        Long count = shoeRepository.countById(id);
        if (count == null || count == 0) {
            throw new UserNotFoundException("Could not find users with
ID " + id);
        }
        shoeRepository.deleteById(id);
    }

    public List<ShoeEntity> searchResult(String shoeCategory) {
        if (shoeCategory != null) {
```

```

        return shoeRepository.searchByCategory(shoeCategory);
    }
    return (List<ShoeEntity>) shoeRepository.findAll();
}
}

```

Properties

```

server.port = 8081
spring.datasource.url=jdbc:mysql://localhost:3306/ecommerce
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.properties.hibernate.format_sql=true
spring.application.name=user-app

```

View

index.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>SportyShoes</title>
    <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">

</head>
<!--<body th:style="'background-image:url(' + @{/images/sport.jpg} +
');'">-->
<body>
<div>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <a class="navbar-brand" th:href="@{/}">SportyShoes.com</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse"
        data-target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent" aria-
expanded="false"
        aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

```

```

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
            </ul>
            <form class="form-inline my-2 my-lg-0">
                <a class="btn btn-light my-2 my-sm-0 mr-1"
th:href="@{/}">Home</a>
                <a class="btn btn-light my-2 my-sm-0 mr-1"
th:href="@{/login}">Admin</a>
                <a class="btn btn-light my-2 my-sm-0"
th:href="@{/reports}">Reports</a>
            </form>
        </div>
    </nav>
</div>
<div class="container">
<div class="jumbotron text-center">
    <h1>Welcome to Sporty Shoes</h1>

    <h4>About</h4>
    <p>Sporty Shoes is a company that manufactures and sells sports
shoes. They have a walk-in store, and now, they wish to launch their
e-commerce portal sportyshoes.com.</p>
</div>
    <br/>
<div class="container">
    <div class="row">
        <div class="col-sm-8">
            <h2>Developed by Dakamanbha Ryngkhlem</h2>
            <h3>Project for Phase-3</h3>
            <p>Under the supervision of <strong
style="color:blue">Simpl<sub>i</sub>Learn</strong></p>
        </div>
        <div class="col-sm-4">
            
        </div>
    </div>
</div>
</div>
</body>
</html>

```

product.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">

```



```

<title>Manage Product</title>
<link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

<div class="container-fluid text-center">
  <div class="h1">Manage Products</div>
  <div class="mr-2">
    <a class = "h3" th:href="@{/shoeDetails/new}">Add New
Product</a>
    <div th:if="${message}" class="alert alert-success text-
center">
      [[${message}]]
    </div>
  </div>

  <div>
    <table class="table table-bordered">
      <thead class="th-dark">
        <tr>
          <th>ID</th>
          <th>Date</th>
          <th>Shoe Color</th>
          <th>Shoe Size</th>
          <th>Shoe Category</th>
          <th>Shoe Price</th>
          <th>Quantity</th>
          <th>Action</th>

        </tr>
      </thead>
      <tbody>
        <th:block th:each = "products : ${listShoes}">
          <tr>
            <td>[[${products.id}]]</td>
            <td>[[${products.dateAdded}]]</td>
            <td>[[${products.shoeColor}]]</td>
            <td>[[${products.shoeSize}]]</td>
            <td>[[${products.shoeCategory}]]</td>
            <td>[[${products.shoePrice}]]</td>
            <td>[[${products.availableInStock}]]</td>
            <td>
              <a class="h6 mr-3"
th:href="@{'/shoeDetails/edit/' + ${products.id}}">Update Product</a>
              <a class="h6 "
th:href="@{'/shoeDetails/delete/' + ${products.id}}">Delete</a>

            </td>
          </tr>

```

```

        </th:block>
    </tbody>
</table>
<div class="text-center">
    <button type="button" class="btn btn-secondary m-2"
onclick="view()">Available Products</button>
    <button type="button" class="btn btn-secondary m-2"
onclick="logout()">Back</button>
</div>

</div>

</div>
<script type="text/javascript">
    function view(){
        window.location = "[@{/shoeDetails}]";
    }
    function logout(){
        window.location="[@{/admin}]";
    }
</script>
</body>
</html>

```

search_form.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>[[@{pageTitle}]]</title>
    <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

<div class="container-fluid text-center">
    <div class="h1">[[@{pageTitle}]]</div>
    <form th:action="@{/report/search}" method="post"
        style="max-width:1000px; margin: 0 auto;">
        <div class="border border-secondary rounded p-3">
            <div class="form-group row">

                <div class="col-sm-4">
                    <label class="col-sm-4 col-form-
label">Filter:</label> <input type="text" name="keyword"

```

```

th:value="${keyword}"

class="form-control" required/>
    </div>
    <div class="col-sm-4">
        <button type="submit" class="btn btn-primary m-2">Search</button>
        <button type="submit" class="btn btn-primary m-2"
onclick="view()">Clear</button>
    </div>
</div>
</div>
</form>

<div>
    <table class="table table-bordered">
        <thead class="th-dark">
            <tr>
                <th>ID</th>
                <th>Date Added</th>
                <th>Shoe Color</th>
                <th>Shoe Size</th>
                <th>Shoe Category</th>
                <th>Added Quantity</th>
                <th>Shoe Price</th>

            </tr>
        </thead>
        <tbody>
            <th:block th:each="products : ${listShoes}">
                <tr>
                    <td>[[${products.id}]]</td>
                    <td>[[${products.dateAdded}]]</td>
                    <td>[[${products.shoeColor}]]</td>
                    <td>[[${products.shoeSize}]]</td>
                    <td>[[${products.shoeCategory}]]</td>
                    <td>[[${products.availableInStock}]]</td>
                    <td>[[${products.shoePrice}]]</td>

                </tr>
            </th:block>
        </tbody>
    </table>

</div>
<div class="text-center">
    <button type="button" class="btn btn-secondary m-2"
onclick="logout()">Back</button>
</div>

```

```

</div>
<script type="text/javascript">
    function view(){
        window.location = "[@{/report}]";
    }
    function logout(){
        window.location="[@{/report}]";
    }
</script>
</body>
</html>

```

login_form.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Admin</title>
    <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

<div>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <a class="navbar-brand" th:href="@{/}">SportyShoes.com</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse"
            data-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-
expanded="false"
            aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
</ul>
            <form class="form-inline my-2 my-lg-0">
                <a class="btn btn-light my-2 my-sm-0 mr-1"
th:href="@{/}">Home</a>
                <a class="btn btn-light my-2 my-sm-0 mr-1"
th:href="@{/login}">Admin</a>
                <a class="btn btn-light my-2 my-sm-0"
th:href="@{/report}">Reports</a>
            </form>
        </div>
    </nav>
</div>

```

```

<div class="container-fluid text-center">
  <div class="h2">Admin Login</div>
  <form th:action="@{/admin}" method="post"
    style="max-width:500px; margin: 0 auto;">
    <div class="border border-secondary rounded p-3">
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">E-mail:</label>
        <div class="col-sm-8">
          <input type="email" name="email" class="form-
control" required minlength="8" maxlength="45"/>
        </div>
      <div class="form-group row">
        <label class="col-sm-4 col-form-
label">Password:</label>
        <div class="col-sm-8">
          <input type="password" name="password"
            class="form-control" required
minlength="4" maxlength="45"/>
        </div>
      </div>
      <div class="text-center">
        <button type="submit" class="btn btn-primary m-
2">Login</button>
        <button type="button" class="btn btn-secondary m-
2" onclick="cancelForm()">Back</button>
      </div>
    </div>
  </div>
</form>
</div>
<script type="text/javascript">
  function cancelForm(){
    window.location = "[@{/}]";
  }
</script>
</body>
</html>

```

report.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>[ ${pageTitle} ]</title>
  <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

```

```

<div class="container-fluid text-center">
  <div class="h1">[ ${pageTitle} ]</div>

```

```

<form th:action="@{/report/search}" method="get"
      style="max-width:1000px; margin: 0 auto;">
  <div class="border border-secondary rounded p-3">
    <div class="form-group row">

<!--      <div class="col-sm-4">-->
<!--      <label class="col-sm-4 col-form-label">Date</label>
<input type="date" name="dateAdded" th:value = "${}" class="form-
control" required/>-->
<!--      </div>-->

      <div class="col-sm-4">
        <label class="col-sm-4 col-form-label">Filter:</label>
<input type="text" name="keyword"
      th:value="${keyword}" class="form-control" required />
      </div>
      <div class="col-sm-4">
        <button type="submit" class="btn btn-primary m-
2">Search</button>
        <button type="submit" class="btn btn-primary m-2"
onclick="back()">Clear</button>

      </div>
    </div>
  </div>
</form>

<div>
  <table class="table table-bordered">
    <thead class="th-dark">
      <tr>
        <th>ID</th>
        <th>Date Added</th>
        <th>Shoe Color</th>
        <th>Shoe Size</th>
        <th>Shoe Category</th>
        <th>Quantity</th>
        <th>Shoe Price</th>

      </tr>
    </thead>
    <tbody>
      <th:block th:each = "products : ${listShoes}">
        <tr>
          <td>[[${products.id}]]</td>
          <td>[[${products.dateAdded}]]</td>
          <td>[[${products.shoeColor}]]</td>
          <td>[[${products.shoeSize}]]</td>
          <td>[[${products.shoeCategory}]]</td>
          <td>[[${products.availableInStock}]]</td>

```

```

        <td>[[${products.shoePrice}]]</td>

    </tr>
</th:block>
</tbody>
</table>

</div>
<div class="text-center">
    <button type="button" class="btn btn-secondary m-2"
onclick="window.print()">Print</button>
    <button type="button" class="btn btn-secondary m-2"
onclick="logout()">Back</button>
</div>
</div>
<script type="text/javascript">
    function logout(){
        window.location="[[@{/}]]";
    }
    function back(){
        window.location="[[@{/report}]]";
    }
</script>
</body>
</html>

```

users.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Manage Users</title>
    <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

<div class="container-fluid text-center">
    <div class="h1">Manage Users</div>
    <div class="mr-2">
        <a class = "h3" th:href="@{users/new}">Add New User</a>
        <div th:if="${message}" class="alert alert-success text-
center">
            [[${message}]]
        </div>
    </div>
    <div class="mr-2">
        <a class = "h3" th:href="@{/shoeDetails}">Manage Products</a>
    </div>
    <div>
        <table class="table table-bordered">

```

```

<thead class="th-dark">
<tr>
    <th>ID</th>
    <th>E-mail</th>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Enabled</th>
    <th>Action</th>
</tr>
</thead>
<tbody>
<th:block th:each = "user : ${listUsers}">
    <tr>
        <td>[[${user.id}]]</td>
        <td>[[${user.email}]]</td>
        <td>[[${user.firstName}]]</td>
        <td>[[${user.lastName}]]</td>
        <td>[[${user.enabled}]]</td>
        <td>
            <a class="h6 mr-3" th:href="@{'/users/edit/' +
${user.id}}">Change Password</a>
            <a class="h6 " th:href="@{'/users/delete/' +
${user.id}}">Delete</a>
        </td>
    </tr>
</th:block>
</tbody>
</table>
<div class="text-center">
    <button type="button" class="btn btn-secondary m-2"
onclick="view()">Available Users</button>
    <button type="button" class="btn btn-secondary m-2"
onclick="logout()">Logout</button>
</div>

</div>

</div>
<script type="text/javascript">
    function view(){
        window.location = "[@{/users}]";
    }
    function logout(){
        window.location="[@{/login}]";
    }
</script>
</body>
</html>

```


add_shoes.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>[[${pageTitle}]]</title>
  <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
</head>
<body>

<div class="container-fluid text-center">
  <div class="h2 text-center">[[${pageTitle}]]</div>
  <form th:action="@{/shoeDetails/save}" method="post"
th:object="${shoe}"
  style="max-width:500px; margin: 0 auto;">
    <input type="hidden" th:field="*{id}">
    <div class="border border-secondary rounded p-3">
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">Date</label>
        <div class="col-sm-8">
          <input type="date" th:field="*{dateAdded}" class="form-
control" required/>
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">Shoe Color</label>
        <div class="col-sm-8">
          <input type="text" th:field="*{shoeColor}" class="form-
control" required/>
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">Shoe Size</label>
        <div class="col-sm-8">
          <input type="text" th:field="*{shoeSize}" class="form-
control" required/>
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">Shoe Category</label>
        <div class="col-sm-8">
          <input type="text" th:field="*{shoeCategory}" class="form-
control" required/>
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-4 col-form-label">Shoe Price</label>
        <div class="col-sm-8">
```

```

        <input type="text" th:field="*{shoePrice}" class="form-
control" required/>
    </div>
</div>

    <div class="form-group row">
        <label class="col-sm-4 col-form-label">Add Quantity</label>
        <div class="col-sm-8">
            <input type="text" th:field="*{availableInStock}"
class="form-control" required/>
        </div>
    </div>
    <div class="text-center">
        <button type="submit" class="btn btn-primary m-2">Add</button>
        <button type="button" class="btn btn-secondary m-2"
onclick="cancelForm()">Cancel</button>

    </div>
</div>
</form>
</div>
<script type="text/javascript">
    function cancelForm(){
        window.location = "[@{/shoeDetails}]";
    }
</script>
</body>
</html>

```

[user_form.html](#)

```
<!DOCTYPE html>
```

```
<html lang="en" xmlns:th="http://www.thymeleaf.org">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>[[${pageTitle}]]</title>
```

```
    <link rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/css/bootstrap.min.css}">
```

```
</head>
```

```
<body>
```

```
<div class="container-fluid text-center">
```

```
<div class="h2 text-center">[[${pageTitle}]]</div>
```

```
    <form th:action="@{/users/save}" method="post" th:object="${user}"
```

```
style="max-width:500px; margin: 0 auto;">
    <input type="hidden" th:field="*{id}">
    <div class="border border-secondary rounded p-3">
    <div class="form-group row">
    <label class="col-sm-4 col-form-label">E-mail:</label>
        <div class ="col-sm-8">
            <input type="email" th:field="*{email}" class="form-control"
required minlength="8" maxlength="45"/>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-sm-4 col-form-label">First Name:</label>
        <div class ="col-sm-8">
            <input type="text" th:field="*{firstName}" class="form-
control" required minlength="4" maxlength="45"/>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-sm-4 col-form-label">Last Name:</label>
        <div class ="col-sm-8">
            <input type="text" th:field="*{lastName}" class="form-
control" required minlength="4" maxlength="45"/>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-sm-4 col-form-label">Password:</label>
        <div class ="col-sm-8">
            <input type="password" th:field="*{password}" class="form-
control"required minlength="4" maxlength="15"/>
        </div>
    </div>
```

```
<div class="form-group row">

  <label class="col-sm-4 col-form-label">Confirmed</label>

  <div class ="col-sm-8">

    <input type="checkbox" th:field="*{enabled}"/>

  </div>

</div>

<div class="text-center">

  <button type="submit" class="btn btn-primary m-2">Save</button>

  <button type="button" class="btn btn-secondary m-2"
onclick="cancelForm()">Cancel</button>

</div>

</div>

</form>

</div>

<script type="text/javascript">

  function cancelForm(){

    window.location = "[@{/users}]";

  }

</script>

</body>

</html>
```