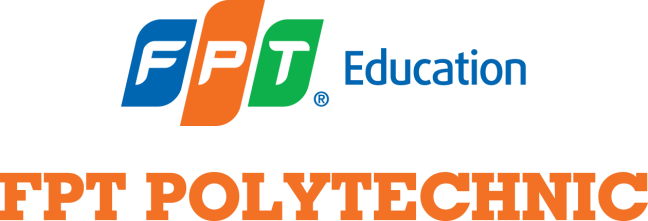
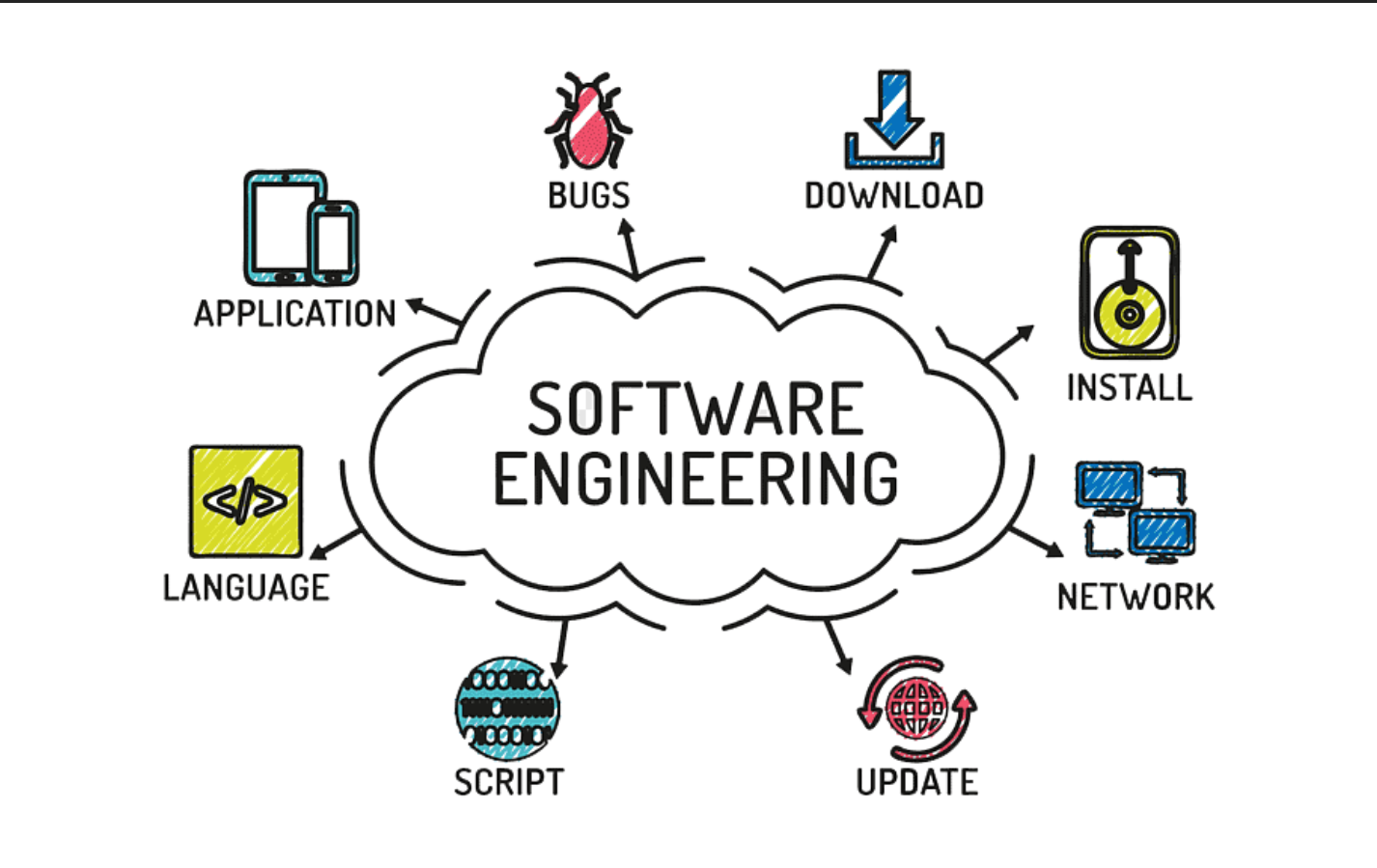
TRƯỜNG CAO ĐẲNG THỰC HÀNH   
FPT POLYTECHNIC TP HCM

**-----\*\*\*-----**



**JAVA 5 - SOF302**

**BÁO CÁO LAB 4**



**GVHD: Thầy Lê Anh Tú**

**Nhóm sinh viên: Nguyễn Xuân Hảo - PS18956  
 Lê Thanh Tú - PS18884  
 Trương Vĩnh Nghi - PS19009  
 Vũ Quốc Thái An - PS17994**

1. **Khởi tạo dự án**

| Tạo mới dự án Spring Boot và thêm các dependencies sau vào POM: |
| --- |
| **Bắt buộc:**   * spring-boot-starter-web (Tạo dự án Spring Web) * spring-boot-start-test (Thêm thư viện test) * tomcat-embed-jasper (Hiển thị trang JSP) * spring-boot-starter-validation (Kiểm lỗi form với Spring)   **Tuỳ chọn:**   * lombok (Hỗ trợ getter/ setter) * spring-boot-devtools (Hỗ trợ reload lại project khi thay đổi) * jstl (Thư viện thẻ Java trong JSP) |
| <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.apache.tomcat.embed</groupId>  <artifactId>tomcat-embed-jasper</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-validation</artifactId>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <optional>true</optional>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <scope>runtime</scope>  <optional>true</optional>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jstl</artifactId>  </dependency>  </dependencies> |

1. **Bài 1 - Tạo paramService**

|  |
| --- |
|  |
|  |
|  |
|  |

* **Tạo SessionService**

|  |
| --- |
|  |

* **Tạo CookiesService**

|  |
| --- |
|  |
|  |

**Bài 2 - Tạo AccountController**

* **AccountController**

| @Controller  @RequestMapping("/account")  public class AccountController {  @Autowired  LoginService loginService;  @GetMapping("/login")  public String login1(Model model) {  User saveUser = loginService.getSaveUser();  model.addAttribute("user", saveUser);  return "login";  }  @PostMapping("/login")  public String login2(Model model) {  boolean result = loginService.login();  model.addAttribute("message", result ? "Login Thành Công!" : "Login Thất Bại");  return "";  }  @RequestMapping("/logout")  public String logout(){  loginService.logout();  return "";  } |
| --- |

* **Mã login.jsp**

| <%@ page contentType="text/html;charset=UTF-8" language="java" %>  <html>  <head>  <title>Title</title>  </head>  <body>  <form action="/account/login" method="post" class="">  <input type="text" name="username">  <input type="password" name="password">  <input type="checkbox" name="remember" value="true">  <button class="">Submit</button>  </form>  </body>  </html> |
| --- |

* **Tiêm ParamService, CookieService vào AccountController để làm việc với tham số va cookies**
* **(username, password, remember) đọc các tham số và ghi nhớ tài khoản đăng nhập được cho login2()**

| @Service  public class LoginService {  @Autowired  ParamService paramService;  @Autowired  CookieService cookieService;  @Autowired  SessionService sessionService;  public boolean login() {  String username = paramService.getString("username", "");  String password = paramService.getString("password", "");  boolean remember = paramService.getBoolean("remember", false);  User user = new User(username, password);  if (checkLoginInfo(user)) {  saveLoginInfo(user, remember);  sessionService.add("user", user);  return true;  }  removeLoginInfo();  return false;  }  public User getSaveUser(){  String username = cookieService.getValue("username");  String password = cookieService.getValue("password");  return username == null ? null : new User(username, password);  }  public void logout() {  sessionService.remove("user");  }  private boolean checkLoginInfo(User user){  return "fpoly".equalsIgnoreCase(user.getUsername()) && "123".equals(user.getPassword());  }  private void saveLoginInfo(User user, boolean remember) {  if (remember) {  int hoursRemember = 10 \* 24;  cookieService.add("username", user.getUsername(), hoursRemember);  cookieService.add("password", user.getPassword(), hoursRemember);  } else {  removeLoginInfo();  }  }  private void removeLoginInfo() {  cookieService.remove("username");  cookieService.remove("password");  } |
| --- |

* **Sau khi chạy**

|  |
| --- |

**Bài 3 - upload hình và sử dụng ParamService.save() để lưu**

**hình.**

**ParamService.save()**

| public File save(MultipartFile file, String folderPath) throws IOException {  String fileName = file.getOriginalFilename();  Path path = Paths.*get*(context.getRealPath(folderPath), fileName);  if (Files.*notExists*(path.getParent())) Files.*createDirectories*(path.getParent());  file.transferTo(path);  return path.toFile();  } |
| --- |

**Bài 4 - interface ShoppingCartService**

| public interface IShoppingCartService {  */\*\**  *\* Thêm mặt hàng vào giỏ hoặc tăng số lượng lên 1 nếu đã tồn tại*  *\* @param id là mã mặt hàng cần thêm*  *\* @return mặt hàng đã được thêm vào hoặc cập nhật số lượng*  *\*/*  Item add(Integer id);  */\*\**  *\* Xóa mặt hàng khỏi giỏ*  *\* @param id mã mặt hàng cần xóa*  *\*/*  void remove(Integer id);  */\*\**  *\* Thay đổi số lượng lên của mặt hàng trong giỏ*  *\* @param id mã mặt hàng*  *\* @param qty số lượng mới*  *\* @return mặt hàng đã được thay đổi số lượng*  *\*/*  Item update(Integer id, int qty);  */\*\**  *\* Xóa sạch các mặt hàng trong giỏ*  *\*/*  void clear();  */\*\**  *\* Lấy tất cả các mặt hàng trong giỏ*  *\*/*  Collection<Item> getItems();  */\*\**  *\* Lấy tổng số lượng các mặt hàng trong giỏ*  *\*/*  int getCount();  */\*\**  *\* Lấy tổng số tiền tất cả các mặt hàng trong giỏ*  *\*\*/*  double getAmount(); |
| --- |

* **Item mô tả mặt hàng**

| @Data  @NoArgsConstructor @AllArgsConstructor  public class Item implements Cloneable{  int id;  String name;  double price;  int quantity;  String img;  String desc;  public static final Map<Integer, Item> *ITEM\_DB* =Map.*of*(  1, new Item(1, "Apple Iphone 13 ProMax", 200, 10, "https://muabandienthoai24h.vn/storage/images/Q1Ad0M0yru\_1648458156.jpg", ""),  2, new Item(2, "Samsung Galaxy 22 Ultra", 200, 10,"https://muabandienthoai24h.vn/storage/images/HbvUO0nHtB\_1648470694.jpg", ""),  3, new Item(3, "Apple Iphone XS Max", 200, 10, "https://muabandienthoai24h.vn/storage/images/9QBW0rWcZH\_1649655018.jpg", ""),  4, new Item(4, "Apple Iphone X", 200, 10, "https://muabandienthoai24h.vn/storage/images/NcLfasHbaa\_1649786551.jpg", ""),  5, new Item(5, "Apple Iphone 12", 200, 10,"https://muabandienthoai24h.vn/storage/images/5ONFwcwNBd\_1648459009.jpg", ""),  6, new Item(6, "Apple Iphone 6 Plus", 200, 10,"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQIXRAz9-S5Qd4A-viMk66wPfbzNAinDIWA32-AE\_a94ayhYURXP5GQeT7InJzyTAZnfCs&usqp=CAU", ""),  7, new Item(7, "Samsung Galaxy 21 Ultra", 200, 10,"https://muabandienthoai24h.vn/storage/images/Zwou8XNGur\_1648479146.jpg", "")  );  public String getDesc(){  return "This is an " + name;  }  @Override  public Object clone() throws CloneNotSupportedException {  return super.clone();  }  } |
| --- |

**Bài 5: Tạo lớp Bean ShoppingCartService**

| @Service  @SessionScope  public class ShoppingCartService implements IShoppingCartService {  Map<Integer, Item> map = new HashMap<>();  Map<Integer, Item> db = Item.*ITEM\_DB*;  @Override  public Item add(Integer id) {  Item item = db.get(id);  return map.merge(id, item, (k, v) -> {  v.setQuantity(v.getQuantity() + 1);  return v;  });  }  @Override  public Item update(Integer id, int qty) {  return map.computeIfPresent(id, (k, v) -> {  v.setQuantity(qty);  return v;  });  }  @Override  public void remove(Integer id) {  map.remove(id);  }  @Override  public void clear() {  map.clear();  }  @Override  public Collection<Item> getItems() {  return map.values();  }  @Override  public int getCount() {  return map.size();  }  @Override  public double getAmount() {  Function<Map.Entry<Integer, Item>, Double>  getItemAmount = (entry) -> getAmount(entry);  return map.entrySet()  .stream()  .map(getItemAmount)  .reduce(Double::*sum*)  .orElse(0d);  }  private double getAmount(Map.Entry<Integer, Item> entry) {  Item item = entry.getValue();  return item.getPrice() \* item.getQuantity();  } |
| --- |

* **Tạo ShoppingCartController**

| @Controller  public class ShoppingCartController {  @Autowired  ShoppingCartService cartService;  @Autowired  SessionService sessionService;  @RequestMapping("/cart/view")  public String viewCart(){  sessionService.add("cartItems", cartService.getItems());  return "item\_index";  }  @PostMapping("/cart/add/{id}")  public String addItem(@PathVariable("id") Integer id){  cartService.add(id);  return "redirect:/cart/view";  }  @PostMapping("/cart/update")  public String updateItem(  @RequestParam("id") Integer id,  @RequestParam("qty") Integer qty  ){  cartService.update(id, qty);  return "redirect:/cart/view";  }  @PostMapping("/cart/remove/{id}")  public String removeItem(@PathVariable("id") Integer id){  cartService.remove(id);  return "redirect:/cart/view";  }  @RequestMapping("/cart/clear")  public String clearCart(){  cartService.clear();  return "redirect:/cart/view";  } |
| --- |

* **Tạo trang JSP**

| <%@ page contentType="text/html;charset=UTF-8" language="java" %>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <html>  <head>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">  <link rel="preconnect" href="https://fonts.googleapis.com">  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">  <script src="https://kit.fontawesome.com/34cc1d8757.js" crossorigin="anonymous"></script>  <script src="https://code.jquery.com/jquery-3.6.1.min.js"  integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ=" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>  <title>Product Landing Page</title>  </head>  <body>  <ul class="">  <c:forEach items="${items}" var="item">  <li class="">${item.name} - $ (${item.price}):  <form action="/cart/add/${item.id}" method="post" class="d-inline-block">  <button class="nav-link">Add to cart</button>  </form>  </li>  </c:forEach>  </ul>  <table class="table table-dark table-striped">  <thead>  <tr>  <th scope="col">#</th>  <th scope="col">Name</th>  <th scope="col">Price</th>  <th scope="col">Quantity</th>  <th scope="col">Amount</th>  <th scope="col"></th>  </tr>  </thead>  <tbody>  <c:forEach items="${cartItems}" var="item">  <tr>  <th scope="row">${item.id}</th>  <td>${item.name}</td>  <td>${item.price}</td>  <td>  <form action="/cart/update" class="" method="post">  <input type="hidden" name="id" value="${item.id}">  <input type="number" class="" name="qty" step="1" min="1" onblur="this.form.submit()" value="${item.quantity}">  </form>  </td>  <td>${item.price \* item.quantity}</td>  <td>  <form action="/cart/remove/${item.id}" class="" method="post">  <button class="nav-link">Remove</button>  </form>  </td>  </tr>  </c:forEach>  </tbody>  </table>  <a href="/cart/clear" class="">Clear Cart</a>  <a href="/items/index" class="">Add More</a>  </body>  </html> |
| --- |

**Result:**

|  |
| --- |

**Add to cart**

|  |
| --- |

**Clear form**

|  |
| --- |