Kokab Askary’s First Full-Stack Web Application

*Project Name# Cakelicious by Aelya*

**Description:**

> This project is about my custom cake busniess. The project is a simple web application:

**## Technologies Used :**

\* mdBootstrap - for front end---HTML/CSS/SASS

\* Javascript/JQuery

\* Build automation tool----Maven --Embedded 3.5.13/1.10.0.20181127-2120

\* MySQL version 8.0.16--- for databse

\* Application framework----Spring MVC

\* Eclipse IDE--- Version: 2018-12 (4.10.0)

\* Programming Language----Core Java & JPA

\* Java--version jre 1.8.0\_181

\* spring-data-jpa---2.0.5.RELEASE

\* Server --🡪The Apache Tomcat---9.0

About the Web application:

Users can browse the index page.

Navbar has the following links:

1. Home
2. About
3. Gallery(Gallery has 3 pages displaying images).
4. Menu
   1. Cakes
   2. Cupcakes
   3. Cookies
   4. Desserts
   5. Order Online
   6. Customize
5. Contact
6. Facebook icon
7. Customer Login icon
8. Cart icon

Footer:

1. Facebook icon
2. Twitter icon
3. Linkden icon
4. Instagram icon
5. Pinterest icon

All links to the social media links to their respective websites

About Page:

Contains an image and about me description.

Gallery Page:

User can browse the 3 Gallery pages using Pagination from MDBootstrap and navigation between pages use Controllers.

Contact Page:

User can fill up Contact form, form validation by Spring annotations.

Login Icon:

Takes user to the login page, if user is not a registered user he can click on registration page link to go to registration page.

Drop-Down Menu:

Cakes link takes user to cake display page, while cupcakes, cookies and dessert pages are under construction.

Order-Online link takes user to the ordering page where user can select the products and add to cart.

Cart functionality is NOT implemented.

Inquiry Link takes user to inquiry form where user can inquire about the customization process, using Create Inquiry function of CRUD

Once user submit the form can see its inquiry details, and can also delete the inquiry. Edit functionality not implemented, reason stated in problems section.

Implementing Delete function of CRUD.

For Admin

Once Users registers , Admin can see the list of all registered customers by” /allCustomer” URL Mapping

Allowing admin to Edit, and Delete the Customers implementing CRUD.

**Project Structure:**

Project Name===CaseStudy-

Entities:

1. ContactInfo.java
2. CustomCake.java
3. Customer.java
4. InquiryCustomer.java
5. LoginUser.Java
6. Orders.java

Controllers:

1. ContactFormController
2. CustomCakeController
3. CustomerOrderController
4. InquiryCustomerController
5. LoginController
6. Navbar Controller

Repository:

1. ContactInformationRepository
2. CustomCakeRepository
3. CustomerOrderRepository
4. InquiryFormRepository
5. CustomerRepository

Service Class/DAO :

1.contactinfo service

* Interface--🡪ContactInfoService

**public** **boolean** saveContactInfo(ContactInfo contactInfo);

**public** **boolean** deleteContactInfoById(Long id);

implements 2 CRUD functions

* Class-🡪 ContactInfoServiceImpl

Implements above metods.

2.customerorder service

* Interface--🡪Customerorder Service

**public** List<Orders> getAllOrders();

**public** Orders getOrderById(Long id);

**public** **boolean** saveOrder(Orders orders);

**public** **boolean** deleteOrderById(Long id);

* Class-🡪 C ustomerorder ServiceImpl

Implements above metods.

3.customerregistration service

* Interface--🡪 CustomerRegistration Service

CRUD Funcationality

**public** List<Customer> getAllCustomers();

**public** Customer getCustomerById(Long id);

**public** **boolean** saveCustomer(Customer customer);

**public** **boolean** deleteCustomerById(Long id);

* Class-🡪 CustomerRegistrationServiceImpl

Implements above metods.

4.inquiry service

* Interface--🡪InquiryService

**public** InquiryForm getInquiryFormById(Long id);

**public** **boolean** saveInquiryForm(InquiryForm inquiryForm);

**public** **boolean** deleteInquiryFormById(Long id);

* Class-🡪 InquiryServiceImpl

Implements above metods.

WEB-Content:

External files :

Css--🡪for css style sheet

Images folder--🡪 images for the application

Js folder--🡪 for external javascripts

Under WEB-INF Folder:

Views folder having all jsp files.

Separate jsp page for navbar , footer and head tags injected in the pages through jsp tags.

Casestudy-dispatcher-servlet.xml:

Main controller of the application

Pom.xml:

Contains all the dependencies required for the project.

Application.properties file:

Contains Hibernate and MySQL properties

**Difficulties:**

Tried to implement Spring security for verifying customer registration and login functions but after implementing all the necessary functionalities my project broke and I was unable to find my wayout.

In order to let the application work had to delete all the functionalities.

While doing the CRUD function for inquiry.jsp the result page was not displaying the total price.

Had to delete default constructor in InquiryForm.java and the application worked but could not implement EDIT functionality as 500 Error was coming error: No default constructor for entity InquiryForm.java

My application was running fine but before submitting when I clicked on Login Icon in the navbar I am getting this error. Tried to resolve it but due to time constraint am not able to solve it at the moment.:

UPDATE

Was able to fix this problem

**Type** Exception Report

**Message** An exception occurred processing [WEB-INF/views/login.jsp] at line [50]

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Exception**

org.apache.jasper.JasperException: An exception occurred processing [WEB-INF/views/login.jsp] at line [50]

47: modelAttribute="loginUser">

48: <div class="form-group md-form pink-text">

49:

50: <form:input type="email" class="form-control deep-purple-text"

51: path="loginUser.email" placeholder="Enter Email" />

52: <label for="Form-email5">Your Email:\*</label>

53: <small>

**Root Cause**

org.springframework.beans.NotReadablePropertyException: Invalid property 'loginUser' of bean class [bnym.casestudy.entity.LoginUser]: Bean property 'loginUser' is not readable or has an invalid getter method: Does the return type of the getter match the parameter type of the setter?

org.springframework.beans.AbstractNestablePropertyAccessor.getPropertyValue(AbstractNestablePropertyAccessor.java:622)

org.springframework.beans.AbstractNestablePropertyAccessor.getNestedPropertyAccessor(AbstractNestablePropertyAccessor.java:839)

org.springframework.beans.AbstractNestablePropertyAccessor.getPropertyAccessorForPropertyPath(AbstractNestablePropertyAccessor.java:816)

org.springframework.beans.AbstractNestablePropertyAccessor.getPropertyValue(AbstractNestablePropertyAccessor.java:610)

org.springframework.validation.AbstractPropertyBindingResult.getActualFieldValue(AbstractPropertyBindingResult.java:104)

org.springframework.validation.AbstractBindingResult.getFieldValue(AbstractBindingResult.java:228)

