Shuqa

Soran Qiji

CST-452 Capstone Project Proposal

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 2.0

Date: April 25th, 2025

ABSTRACT

The Shuqa project, Shuqa is a Chaldean (Aramaic) word for bazar, is a web application that people can use to sell and purchase products locally in areas where bank cards are not widely used. Everyone can access the website with no need for credit/debit cards or any other online payment methods such as PayPal, as cash is the only accepted payment method. And buyers and sellers should arrange the location of the purchased item pick up. The users must register providing their information such as full name, email address, and phone number to create usernames to access the website. No membership or fees are required.

History and Signoff Sheet

Change Record

Date	Author	Revision Notes
09/16/2024	Soran Qiji	Initial draft for review/discussion V 1
04/26/2025		Work Breakdown Structure, Project completion criteria, Assumptions and Constraints, use of technology, Project scope (search for product is in scope), UML, Sitemap, wireframe, ER, Change Control loge.

Overall Instructor Feedback/Comments

Overall Instructor Feedback/Comments

Integrated Instructor Feedback into Project Documentation

☐ Yes ☒ No

Project Approval

☒ Professor Mark Reha

Contents

PROJECT OVERVIEW AND PROJECT OBJECTIVES 4
STATE THE PROBLEM AND BACKGROUND 4
PROJECT OBJECTIVES 4
CHALLENGES 4
BENEFITS AND OPPORTUNITIES 4
PROJECT SCOPE 5
IN SCOPE 5
OUT SCOPE 5
WORK BREAKDOWN STRUCTURE 6
PROJECT SUCCESS MEASURES 8
PROJECT HIGH-LEVEL SOLUTION 9
INTRODUCTION 9
USE OF TECHNOLOGY: 9
SITEMAP: 10
UML: 11
ER DIAGRAM: 12
WIREFRAME: 12
Home page: *12*
Product page: *13*
Display single product: *13*
My Product page: *14*
Create item page: *14*
Update item page: *15*
Delete Item: *15*
Forget Password page: *16*
Reset Password Page: *16*
Login Page: *17*
Register page: *17*
PSEUDOCODE: 18
PROJECT CONTROLS 18
RISK MANAGEMENT AND ISSUES LOG: 18
CHANGE CONTROL LOGE: 19
ROLES AND RESPONSIBILITIES 19
PROJECT COST AND SCHEDULE 20
APPENDIX A – REFERENCES 21
APPENDIX B – COPYRIGHT COMPLIANCE 21

Project Overview and Project Objectives

State the Problem and Background

People usually struggle to conduct online marketing business in areas where bank cards and other online payment methods, such as PayPal, are not widely accessible. Having a website that allows people to search for their needs in the comfort of their homes and conduct transactions using their preferred payment methods, paying with cash in this case, will reduce problems and ease their life.

Project Objectives

- The ability of user to visit the website and explore the listed items.
- The ability of user to register to the website by providing personal information like name, email, phone number ….
- The ability of user to log in using the username and password.
- The ability of user to create posts by providing item information.
- The ability of user to update and delete the post

Challenges

- The ability to check the users' authorization by comparing the username and password that they provided with the one in database.
- The ability to give permission to the users to update or delete only the item that they posted.

Benefits and Opportunities

Shuqa website will provide business and marketing opportunities for people with limited payment options to pay for their merchandise. The website will take advantage of the locals living in towns and small cities where their financial communication does not depend on bank transactions. In cities where people use cash only to manage their lives and pay for their daily needs. The website will establish trading possibilities for people who have access to cash only and cannot use alternative paying options.

Project Scope

In Scope

- Home page: Users can get information about the website.
- Log in page: The users can login to the account by entering their username and password.
- Register page: The users can create an account.
- Products page:
    i. The users can list their products to sell.
    ii. The users can browse the page for purchase purposes.
- Check the users' authorization by comparing the username and password with the one on database.
- The users will be able to update or delete the post.
- The user will be able to search

Out Scope

- Admin page:
    i. Create a table for all users with passwords
    ii. Create a table for the listed item.
    iii. The ability to modify the post or remove it.
- Showing the item on the map.
- Chat/message between buyer and seller.
- Feedback.

2.

Stakeholder Name	Role(s)	Responsibilities
Soran Qiji	Owner	Project manager
	Developer	Planning and develop the web-application

Work Breakdown Structure

Work Breakdown Structure										
ID	Task	Dependencies	Status	Effort Hours	Cost	Start Date	Planned Completion	Estimate to Completion	Actual Completion	Resource
1	Home page	Home page with about the website	Done	4		11/20/24			11/25/24	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
2	Log in page	Where the user can log in using username and password	Done	4		12/1/24			12/2/24	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
3	Register page	Where user can register	Done	8		11/26/24			12/1/24	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
4	Products page	Where customers can see products	Done	8		11/24/24			1/20/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
5	My Products page	Where users can see their items	Done	6		1/22/25			1/23/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
6	Create Product page	Where user can post new item	Done	8		1/24/25			1/28/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
7	Update Product page	Where user can update an item	Done	8		2/8/25			2/11/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
8	Delete Product	Where user can delete a product	Done	3		2/11/25			2/12/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
9	Display single product page	Where user can display a single item	Done	5		1/21/25			1/22/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS
10	Rest password pages	Where user can change password	Done	10		1/30/25			2/8/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS, Gmail
11	Database	To store data	Done	2		11/20/24			11/20/24	MongoDB
12	Cloud	To store code	Done	1		11/20/24			11/20/24	GitHub

13	Search for product	User can search for product	Done	7		2/26/25			3/1/25	VS code, Spring Boot, Thymeleaf, Bootstrap, CSS

Project Success Measures

1. The ability of users to register and login to the website, posting products for sell and the ability to edit and delete the posted item will be the measurement of project success.

Project completion criteria

Project Completion Criteria
1 - Having home page that describe the website.
2 - Having a registration page.
3- Having login page
4- Having product page to browse
5- Having single product page
6- User can create post
7- User can update post
8- User can delete post
9- User can change password
10- User can search for a product

Assumptions and Constraints

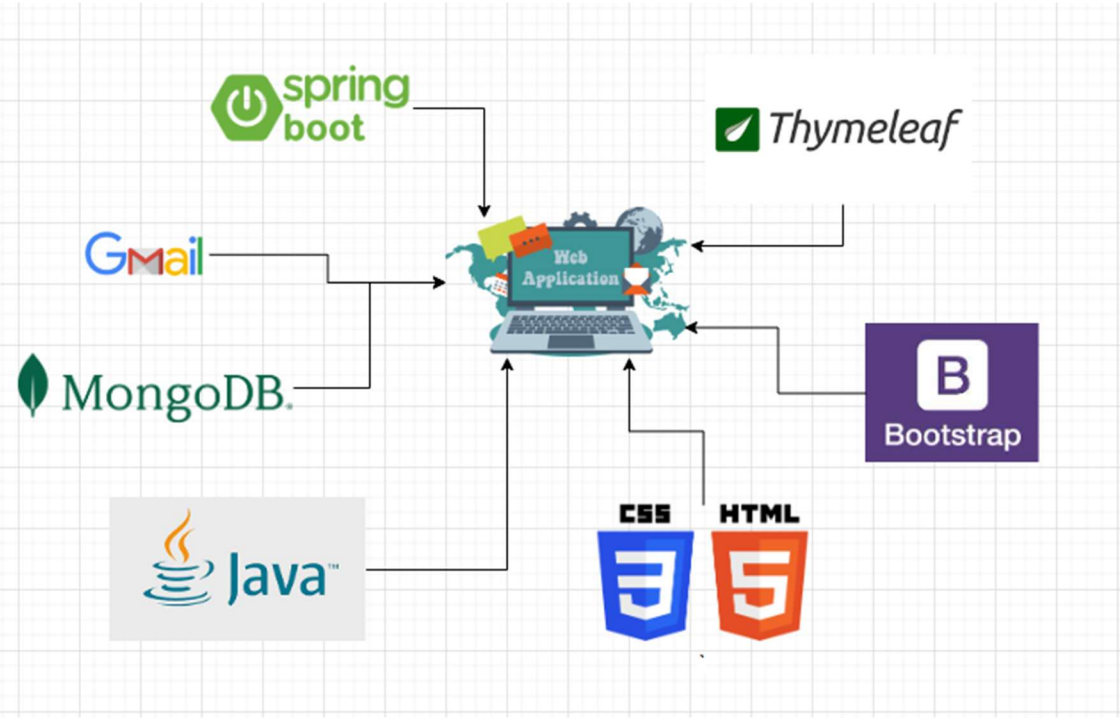
Assumptions and Constraints					
ID	Description	Comments	Type	Status	Date Entered
1	Spring Boot	While the language used to build the project is Java, Spring Boot framework was decided to be used	Assumption	Done	09/20/24
2	Thymeleaf	Java HTML templet	Assumption	Done	09/20/24
3	Bootstrap with CSS	CSS framework for styling the website	Assumption	Done	09/20/24
4	User authorization	The ability to check the users' authorization and link the user with the post that they added, therefore, they can edit and delete only the post that they added	Constraints	Done	11/26/24
5	Image upload	The ability to upload images with image format only	Constraints	Done	2/16/25
	Password Change	Then user can change the password	Constraints	Done	1/30/25
	Search for product	User can search for product	Constraints	Done	2/26/2025

Project High-Level Solution

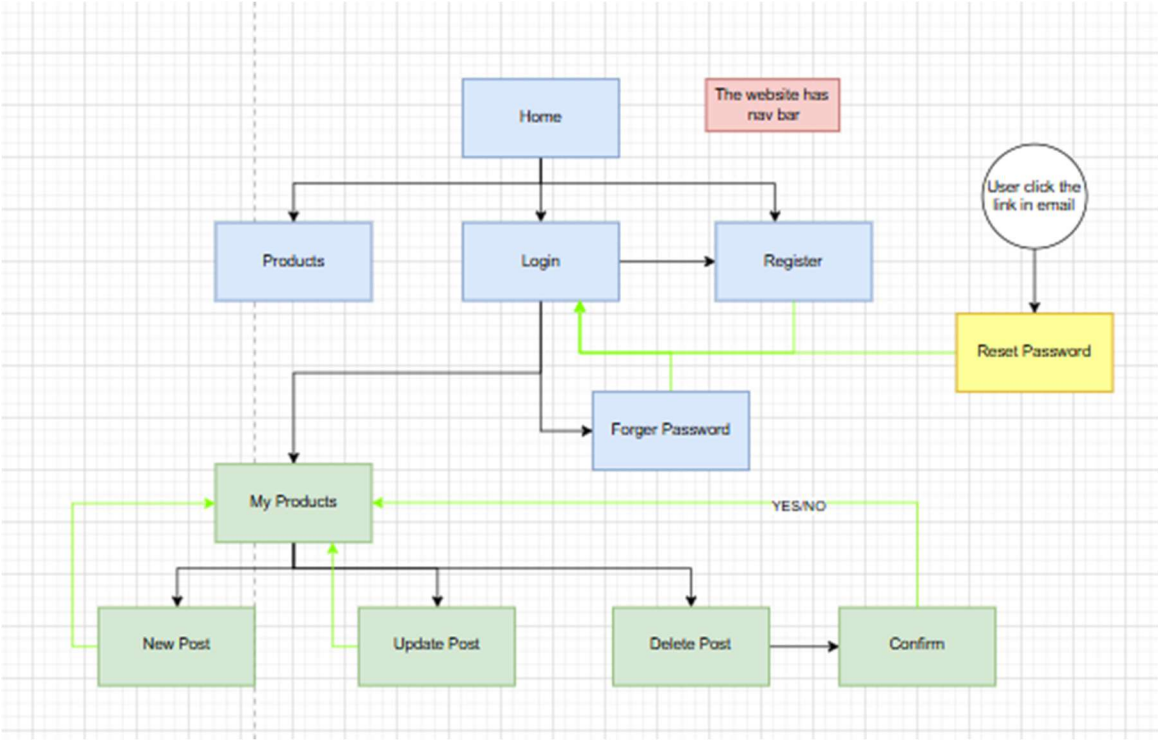
Introduction

As mentioned earlier, this website will be one of the kind business opportunities in areas where people don't rely on banking transactions and wire transfers for managing and paying for their daily lives' requirements. Some of the challenges this website may face are people developing faith in it, since it will be the first of its kind in the field and starting to use it to conduct and establish their business. The success of this website will mainly depend on small businesses and individuals who want to start a trustworthy and prosperous business and their engagement with their customers and their ability to maintain a community of dedicated clients.

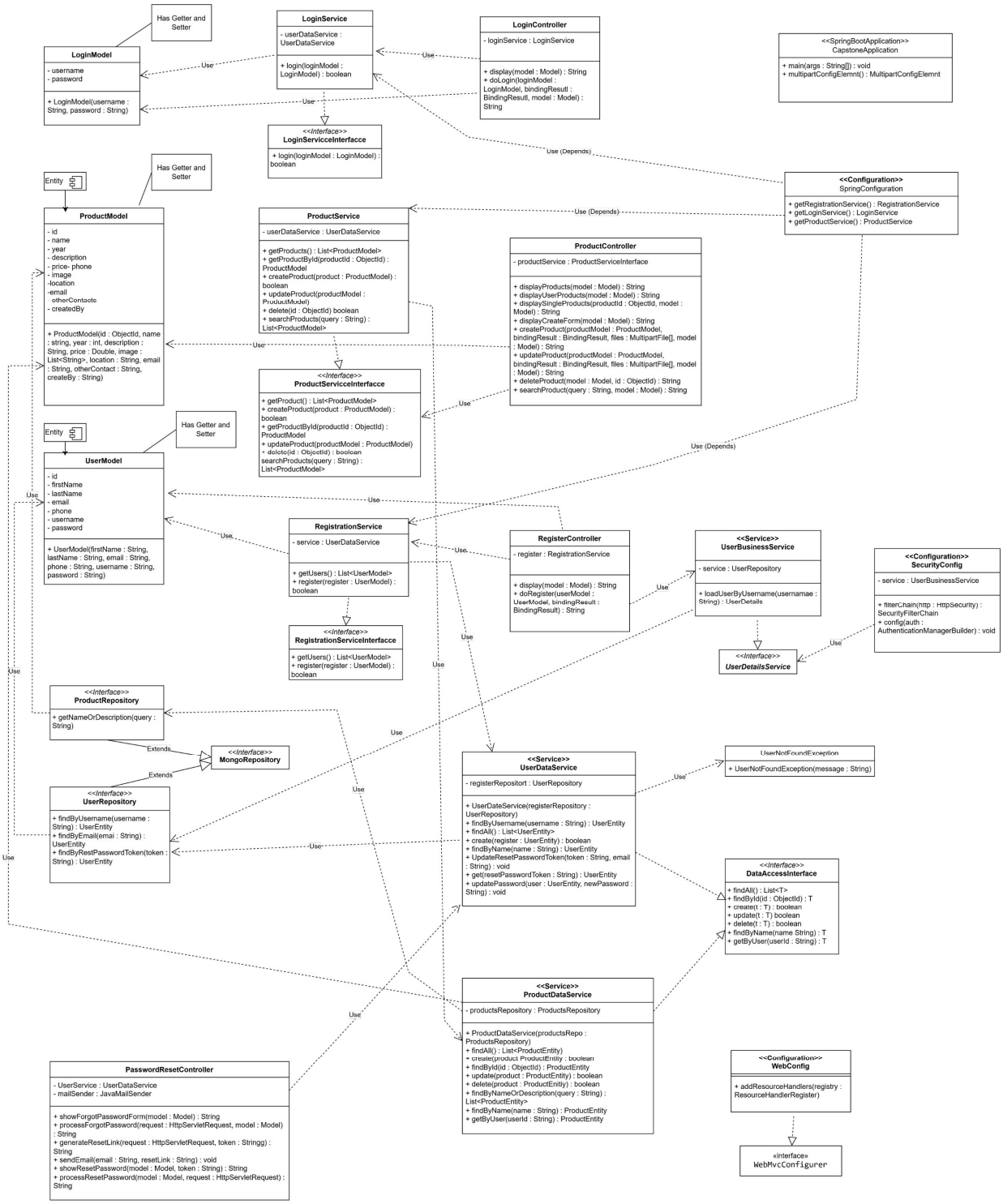
Use of Technology:



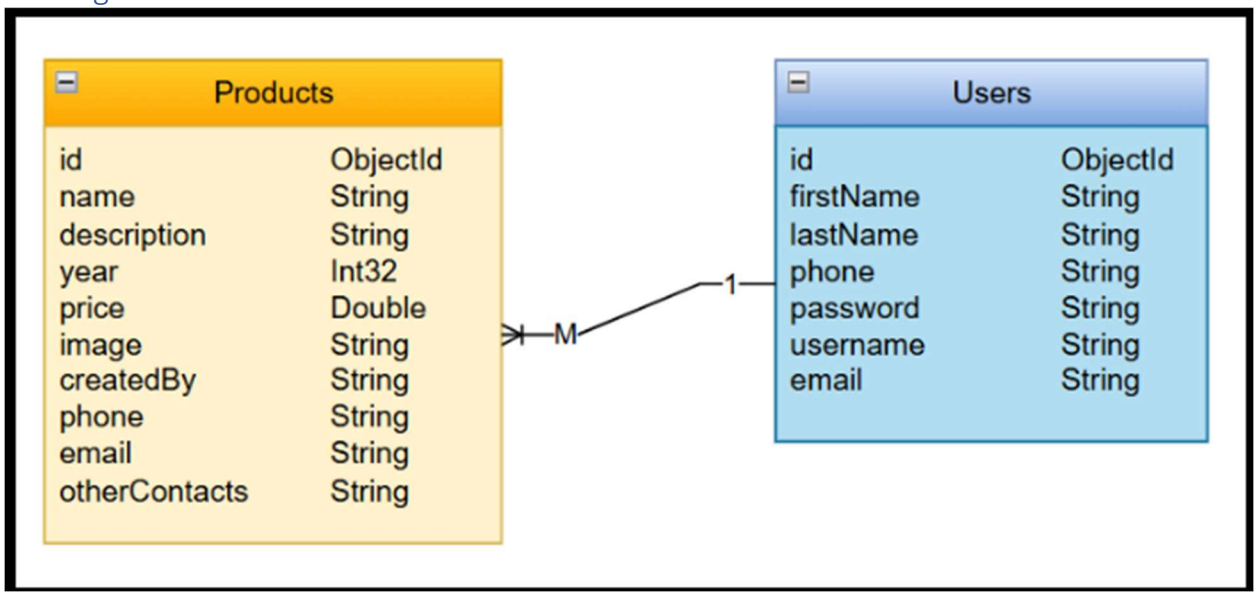
SiteMap:


Home
The website has nav bar
User click the link in email
Products
Login
Register
Reset Password
Forger Password
My Products
YES/NO
New Post
Update Post
Delete Post
Confirm

UML:

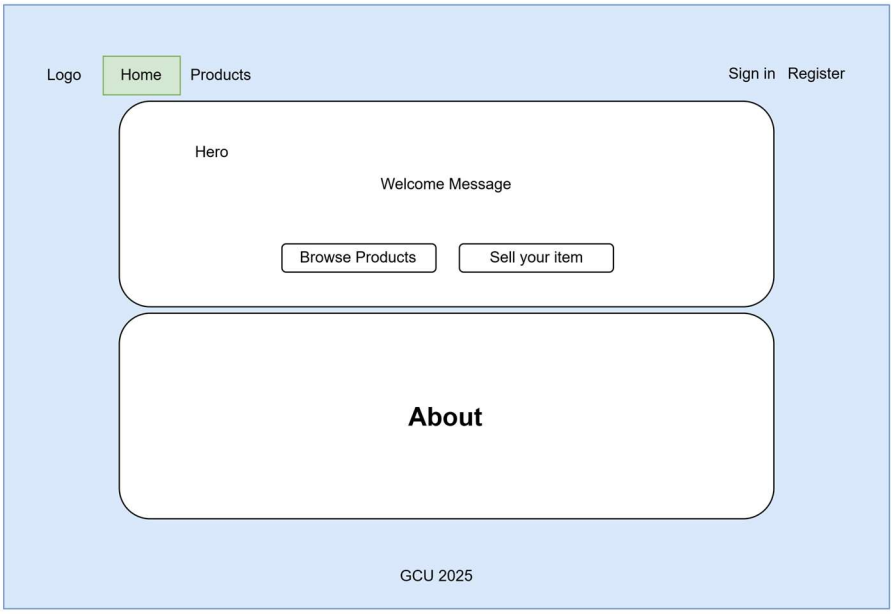

Has Getter and Setter
LoginModel
- username
- password
+ LoginModel(username : String, password : String)
Use
LoginService
- userDataService : UserDataService
+ login(loginModel : LoginModel) : boolean
Use
LoginController
- loginService : LoginService
+ display(model : Model) : String
+ doLogin(loginModel : LoginModel, bindingResutl : BindingResutl, model : Model) : String
<<SpringBootApplication>>
CapstoneApplication
+ main(args : String[]) : void
+ multipartConfigElemnt() : MultipartConfigElemnt
Use
<<Interface>>
LoginServicceInterfacce
+ login(loginModel : LoginModel) : boolean
Use (Depends)
Entity
Has Getter and Setter
ProductModel
- id
- name
- year
- description
- price- phone
- image
-location
-email
-otherContacts
- createdBy
+ ProductModel(id : ObjectId, name : string, year : int, description : String, price : Double, image : List<String>, location : String, email : String, otherContact : String, createBy : String)
ProductService
- userDataService : UserDataService
+ getProducts() : List<ProductModel>
+ getProductById(productId : ObjectId) : ProductModel
+ createProduct(product : ProductModel) : boolean
+ updateProduct(productModel : ProductModel)
+ delete(id : ObjectId) boolean
+ searchProducts(query : String) : List<ProductModel>
Use (Depends)
<<Configuration>>
SpringConfiguration
+ getRegistrationService() : RegistrationService
+ getLoginService() : LoginService
+ getProductService() : ProductService
ProductController
- productService : ProductServiceInterface
+ displayProducts(model : Model) : String
+ displayUserProducts(model : Model) : String
+ displaySingleProducts(productId : ObjectId, model : Model) : String
+ displayCreateForm(model : Model) : String
+ createProduct(productModel : ProductModel, bindingResult : BindingResult, files : MultipartFile[], model : Model) : String
+ updateProduct(productModel : ProductModel, bindingResult : BindingResult, files : MultipartFile[], model : Model) : String
+ deleteProduct(model : Model, id : ObjectId) : String
+ searchProduct(query : String, model : Model) : String
Use
<<Interface>>
ProductServicceInterfacce
+ getProduct() : List<ProductModel>
+ createProduct(product : ProductModel) : boolean
+ getProductById(productId : ObjectId) : ProductModel
+ updateProduct(productModel : ProductModel)
+ delete(id : ObjectId) : boolean
searchProducts(query : String) : List<ProductModel>
Use (Depends)
Entity
Has Getter and Setter
UserModel
- id
- firstName
- lastName
- email
- phone
- username
- password
+ UserModel(firstName : String, lastName : String, email : String, phone : String, username : String, password : String)
Use
RegistrationService
- service : UserDataService
+ getUsers() : List<UserModel>
+ register(register : UserModel) : boolean
Use
RegisterController
- register : RegistrationService
+ display(model : Model) : String
+ doRegister(userModel : UserModel, bindingResult : BindingResult) : String
Use
<<Service>>
UserBusinessService
- service : UserRepository
+ loadUserByUsername(usernamae : String) : UserDetails
<<Configuration>>
SecurityConfig
- service : UserBusinessService
+ filterChain(http : HttpSecurity) : SecurityFilterChain
+ config(auth : AuthenticationManagerBuilder) : void
Use
<<Interface>>
RegistrationServiceInterfacce
+ getUsers() : List<UserModel>
+ register(register : UserModel) : boolean
Use
<<Interface>>
UserDetailsService
Use
<<Interface>>
ProductRepository
+ getNameOrDescription(query : String)
Extends
<<Interface>>
MongoRepository
Extends
<<Interface>>
UserRepository
+ findByUsername(username : String) : UserEntity
+ findByEmail(email : String) : UserEntity
+ findByRestPasswordToken(token : String) : UserEntity
Use
<<Service>>
UserDataService
- registerRepositort : UserRepository
+ UserDateService(registerRepository : UserRepository)
+ findByUsername(username : String) : UserEntity
+ findAll() : List<UserEntity>
+ create(register : UserEntity) : boolean
+ findByName(name : String) : UserEntity
+ UpdateResetPasswordToken(token : String, email : String) : void
+ get(resetPasswordToken : String) : UserEntity
+ updatePassword(user : UserEntity, newPassword : String) : void
Use
UserNotFoundException
+ UserNotFoundException(message : String)
Use
<<Interface>>
DataAccessInterface
+ findAll() : List<T>
+ findById(id : ObjectId) : T
+ create(t : T) : boolean
+ update(t : T) boolean
+ delete(t : T) : boolean
+ findByName(name String) : T
+ getByUser(userId : String) : T
Use
<<Service>>
ProductDataService
- productsRepository : ProductsRepository
+ ProductDataService(productsRepo : ProductsRepository)
+ findAll() : List<ProductEntity)
+ create(product ProductEntity : boolean
+ findById(id : ObjectId) : ProductEntity
+ update(product : ProductEntity) : boolean
+ delete(product : ProductEnitiy) : boolean
+ findByNameOrDescription(query : String) : List<ProductEntity>
+ findByName(name : String) : ProductEntity
+ getByUser(userId : String) : ProductEntity
Use
PasswordResetController
- UserService : UserDataService
- mailSender : JavaMailSender
+ showForgotPasswordForm(model : Model) : String
+ processForgotPassword(request : HttpServletRequest, model : Model) : String
+ generateResetLink(request : HttpServletRequest, token : Stringg) : String
+ sendEmail(email : String, resetLink : String) : void
+ showResetPassword(model : Model, token : String) : String
+ processResetPassword(model : Model, request : HttpServletRequest) : String
<<Configuration>>
WebConfig
+ addResourceHandlers(registry : ResourceHandlerRegister)
«interface»
WebMvcConfigurer

ER Diagram:

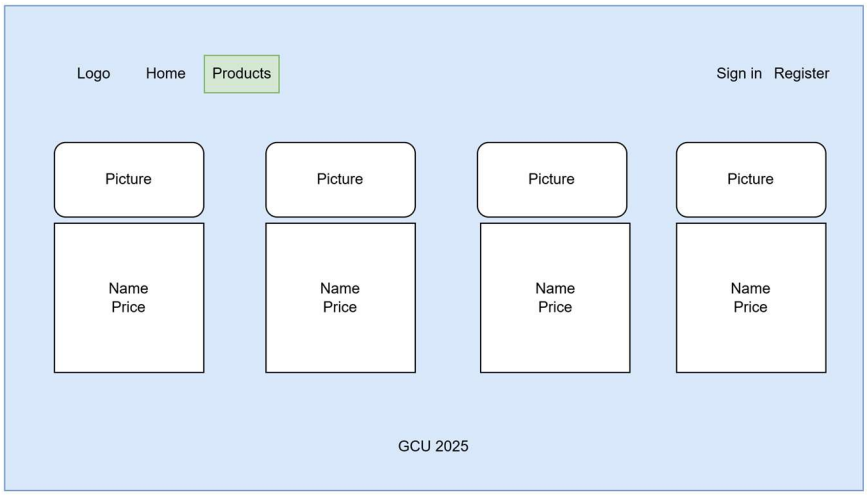


Wireframe:

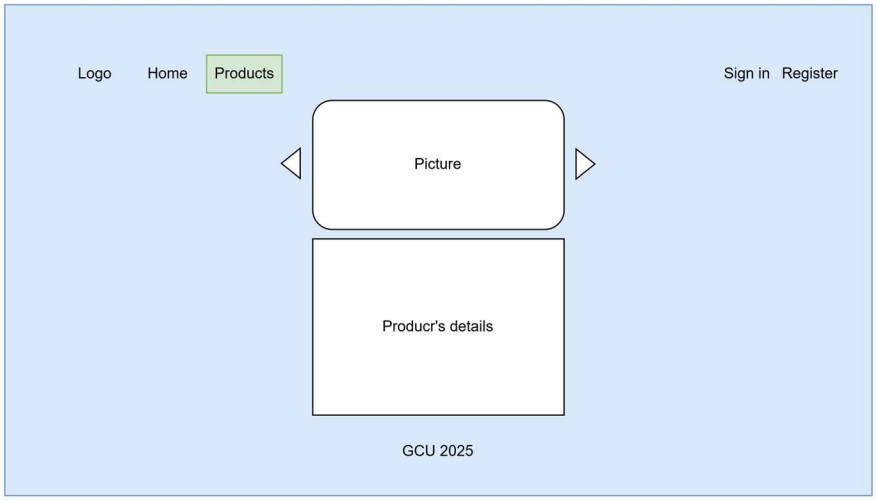
Home page:



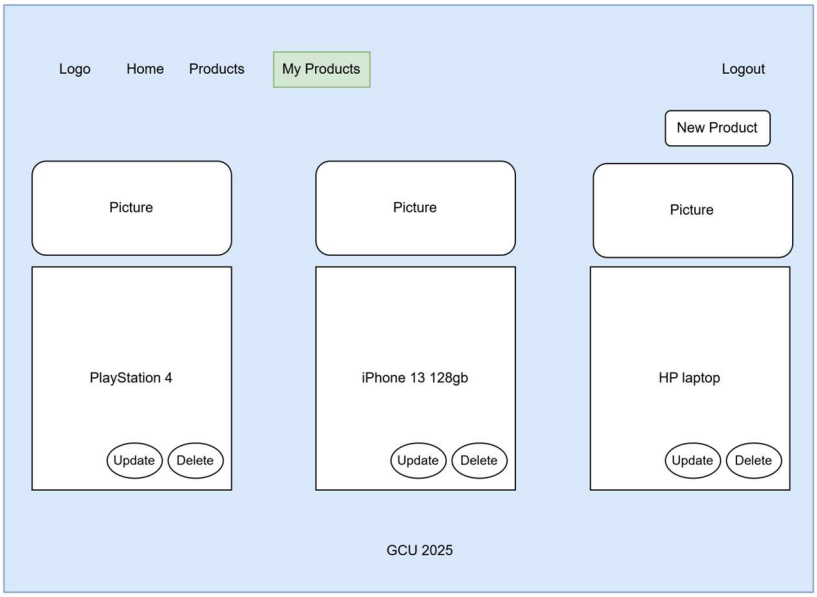
Product page:



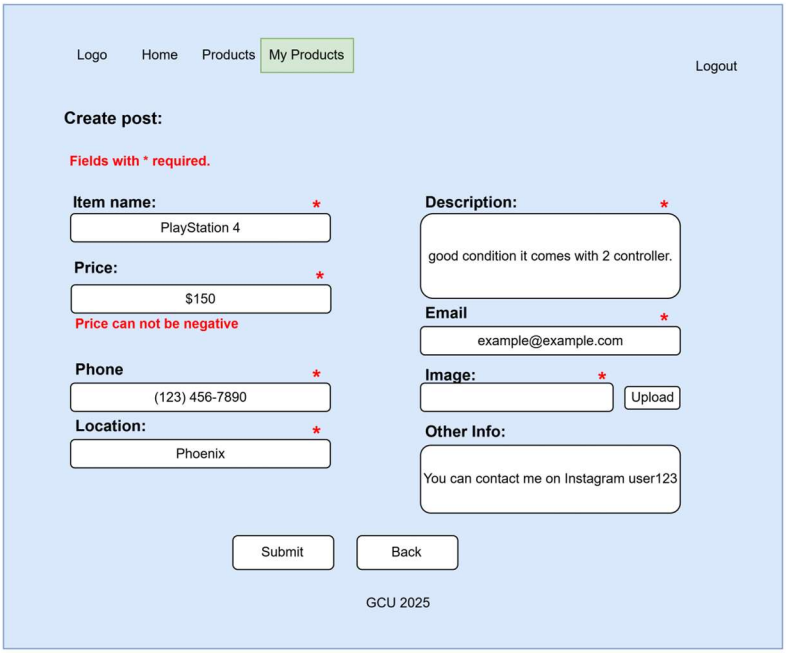
Display single product:



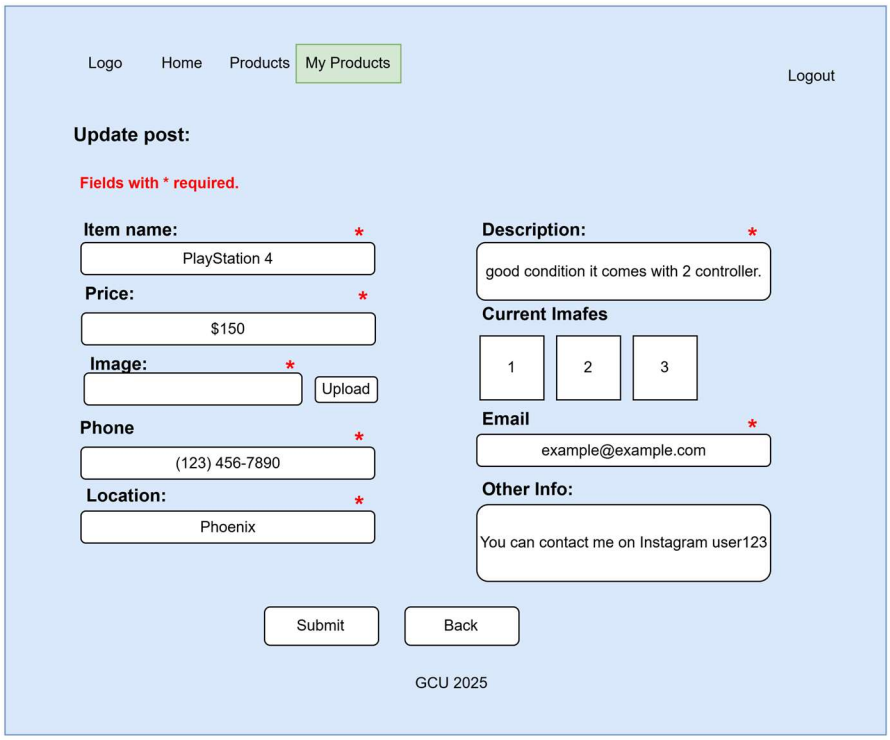
My Product page:



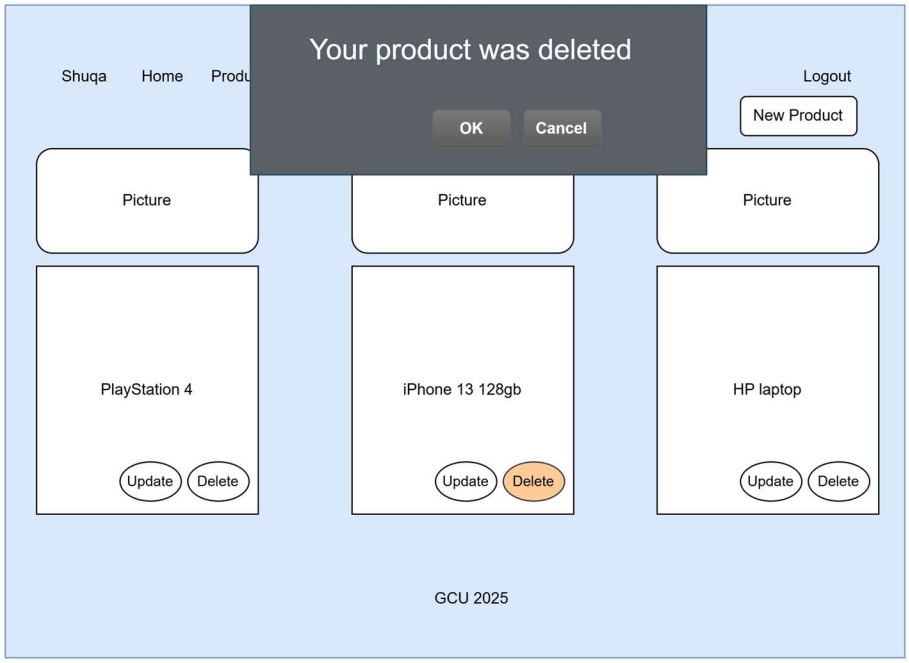
Create item page:



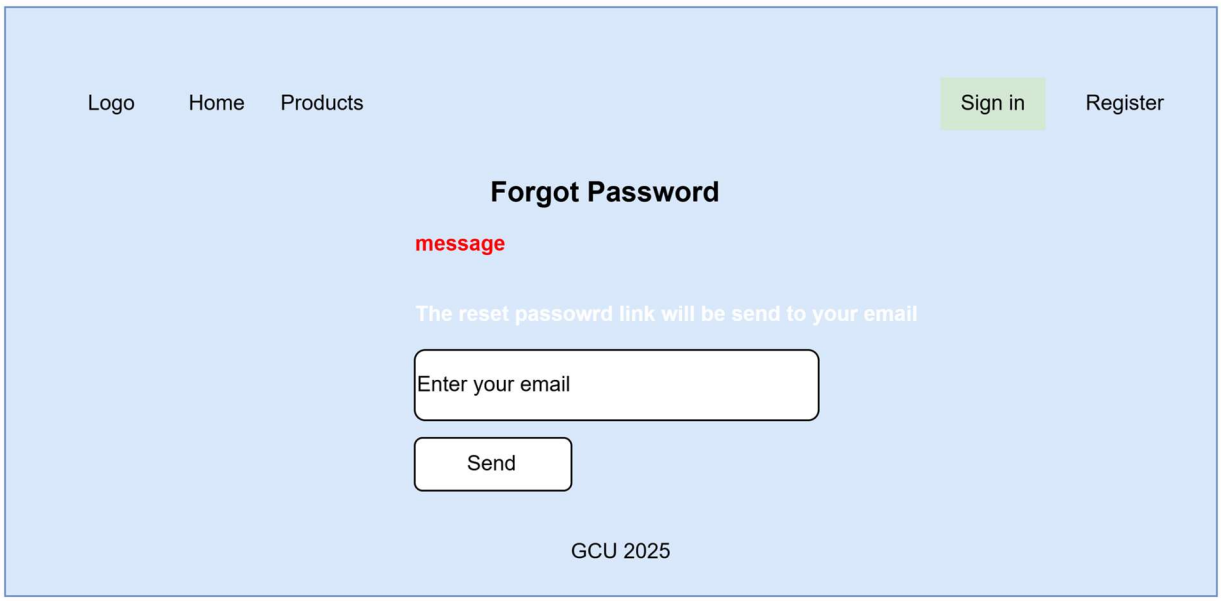
Update item page:



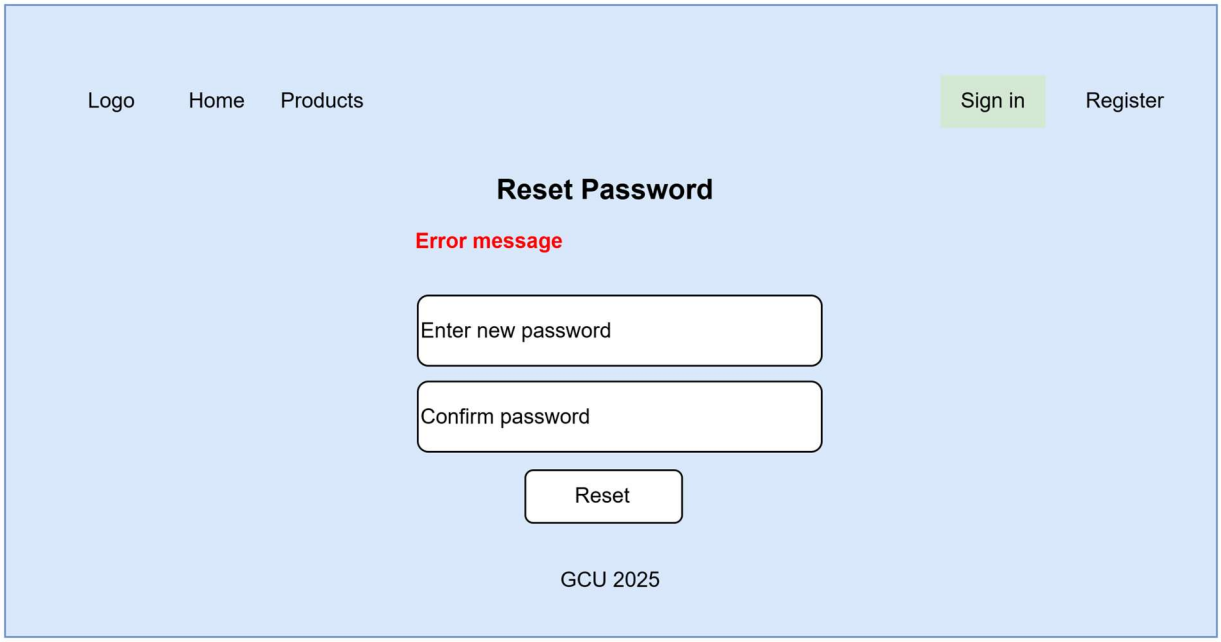
Delete Item:



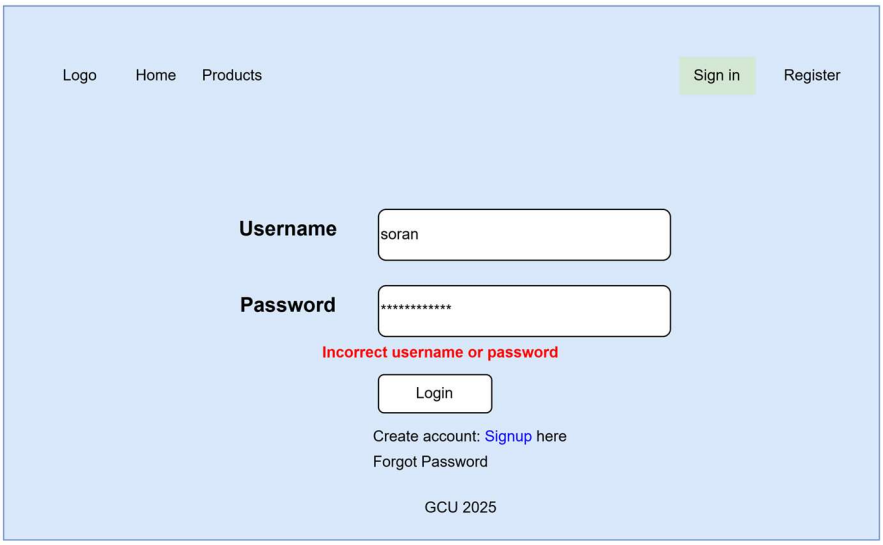
Forget Password page:



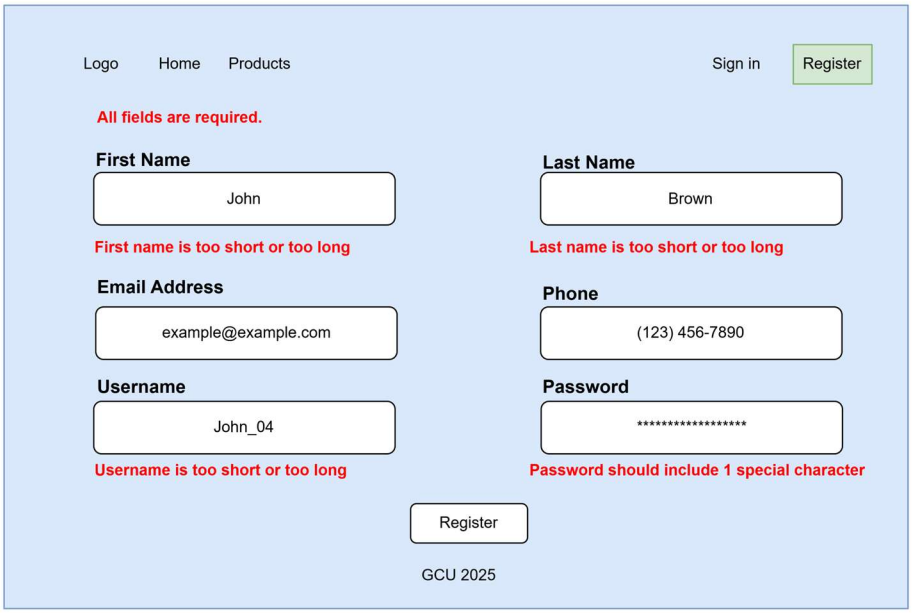
Reset Password Page:



Login Page:



Register page:



Pseudocode:

Enter URL >> Home page displayed >>

Click product page >> list all the item

Click a product >> display product details

Click add new item >> user logged in

Yes >> display add new item page

No >> navigate to login page

Log in >> enter username and password >> user available

No >> navigate to register page

User registration >> enter user information >> submit (create user) >>

navigate to login page.

Yes >> navigate to product page

Display the list of items (products) with ability to edit and remove post (item)

Click forgot password >> enter valid email >> email send

Click the link in email >> enter new password >> submit.

Project Controls

Risk Management and Issues Log:

Risk Management				
Event Risk	**Risk Probability (high, medium, low)**	**Risk Impact**	**Risk Mitigation**	**Contingency Plan**
What is the risk?	What is the probability?	What is the impact if the risk occurs?	What can be done to minimize the risk?	What can be done to minimize the impact of the risk?

Issues Log

ID	Description	Project Impact	Action Plan/Resolution	Owner	Importance	Date Entered	Date to Review	Date Resolved
1	What is the issue?	How will this impact scope, schedule & cost?	How do you intend to deal with this issue?	Who manages this issue?				
2								
3								

Change Control loge:

Change Control Log									
ID	Change Description	Priority	Originator	Date Entered	Date Assigned	Evaluator	Status	Date of Decision	Included in Rev. #
1	Adding search feature	In scop	Soran Qiji	09/21/24	2/26/25	Soran Qiji	Done	2/26/25	4
2	Admin page	Out of scope	Soran Qiji	09/21/24	09/21/24	Soran Qiji	Not Stated		1.0

Roles and responsibilities

Roles and Responsibilities			
Name	Team	Project Role	Responsibility
Soran Qiji	- Planning and design - Development - Testing	Owner	- Manage the project - Developing the project (web app) - Testing the project (web app)

Project Cost and Schedule

1- The project will not require any cost calculations.

2- Project Schedule

Week 1 - 2	Project proposal and Database
Week 3 – 5	Create home and login page
Week 6 – 8	Create register page and working on user authorization
Week 9 – 11	Create, update and delete product
Week 12 - 14	Image upload and searching for product

With each face (week) there was testing and maintenance.

Appendix A – References

List all references using APA style

Appendix B – Copyright Compliance

The technical tools that are used to build the web application are open source such as Eclipse, Java language, Spring Boot, Thymeleaf, MongoDB, and Bootstrap. copyright.gov was used to check for copyright for website name (shuqa).