Assignment - 2

|  |  |
| --- | --- |
| **Student Name/ID Number:** | Huynh Minh Phu - PGSE0823 |
| **Unit Number and Title:** | Module 4 – Web Development Foundations |
| **Academic Year:** | 2023 |
| **Unit Assessor:** | Module 4 |
| **Project Title:** | Setup Spring project, Develop Models & Views. |
| **Issue Date:** | 19/03/2024 |
| **Submission Date:** | 19/03/2024 |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: |

|  |
| --- |
| **Purpose of this project** |
| **Purpose of this project**  To demonstrate your capabilities in the following areas:   * Create Models & Views |
| **Submission Format** |
| 1. Screen capture of Views 2. Sources of Models |
| **Project Brief & Guidance** |
| **Scenario:**  **Refer to the Project Scenario for the Module Project**  You currently work as a *Data Engineer* for Brightica design agency, where you design and implement data models for client-centric products. As part of the role, your manager Mr. Andrew assigned the project to develop an optimal database design to deliver Rich Internet Application for boutique. Boutique is a marketplace for sellers to promote their products and for consumers to purchase with ease. The company wants to have a consumer-centric application with an enhanced user experience.  The scope of the project in this module is for development of Spring Website.  **The overview of the project is as below**  The overview of the project is as below There are 3 types of users   1. Sellers 2. Consumers 3. Administrator   Sellers should be able to perform following functions in the portal   1. Register in the portal 2. Update their Profile after logging in 3. Maintain the product catalog to promote their products   Consumers should be able to perform following functions in the portal   1. Register in the portal 2. Update their Profile after logging in 3. Search products 4. Choose products to view the details 5. Shopping cart to add, review, and remove items |

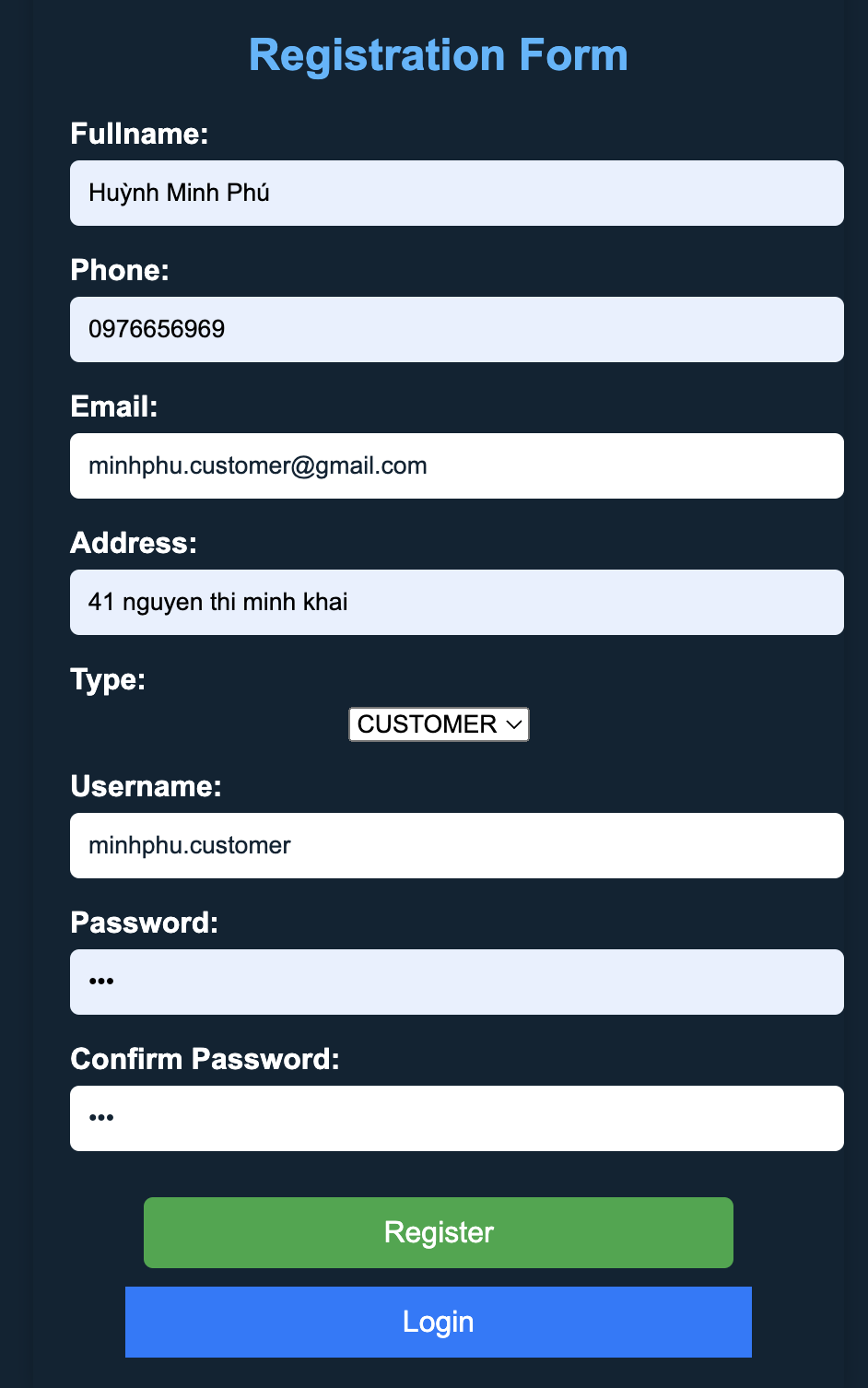
|  |
| --- |
| Administrator should be able to perform following functions in the portal   1. Administer user data. 2. Send bulk email invite to potential clients to register   **The scope of this assignment**  The scope is to setup a *Spring Project*, Create all the Spring Forms for the following functionality.  *Consumers* should be able to perform following functions in the portal   1. Allow the consumers to register in the portal, show a thank you page & send a registration confirmation email. 2. Search & Find Products after login and view their profile. 3. Provide Login Page 4. Provide password retrieval functionality. 5. Update their Profile after logging in.   *Sellers* should be able to perform following functions in the portal   1. Allow the sellers to register in the portal, show a thank you page & send a registration confirmation email. 2. Maintain the product catalogue after login and view their profile. 3. Provide Login Page 4. Provide password retrieval functionality. 5. Update their Profile after logging in.   The site does not need a backend integration (Back end integration will be done in Mini Project 3), but all other display views, java script validations, url mappings etc.   * 1. Provide screen capture of HTML Pages with Spring forms.   2. Provide sources of Models |

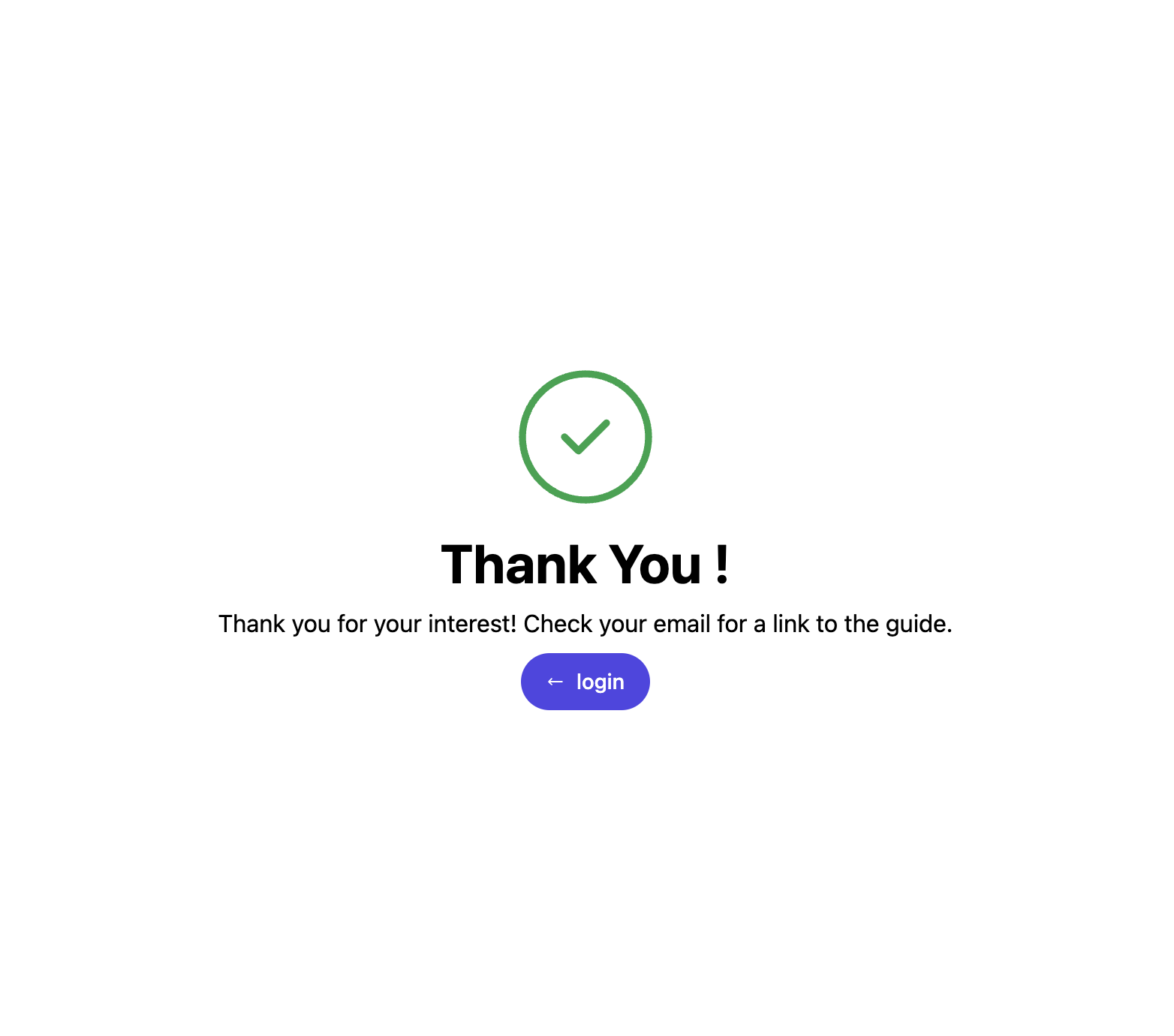
**The scope of this assignment**

The scope is to setup a *Spring Project*, Create all the Spring Forms for the following functionality.

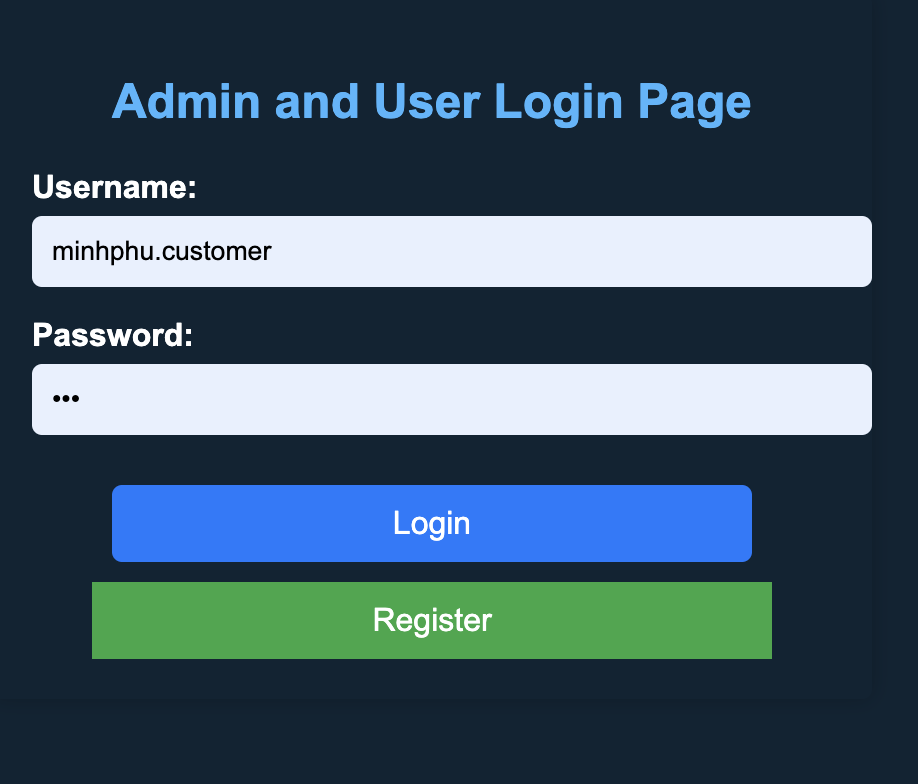
*Consumers* should be able to perform following functions in the portal

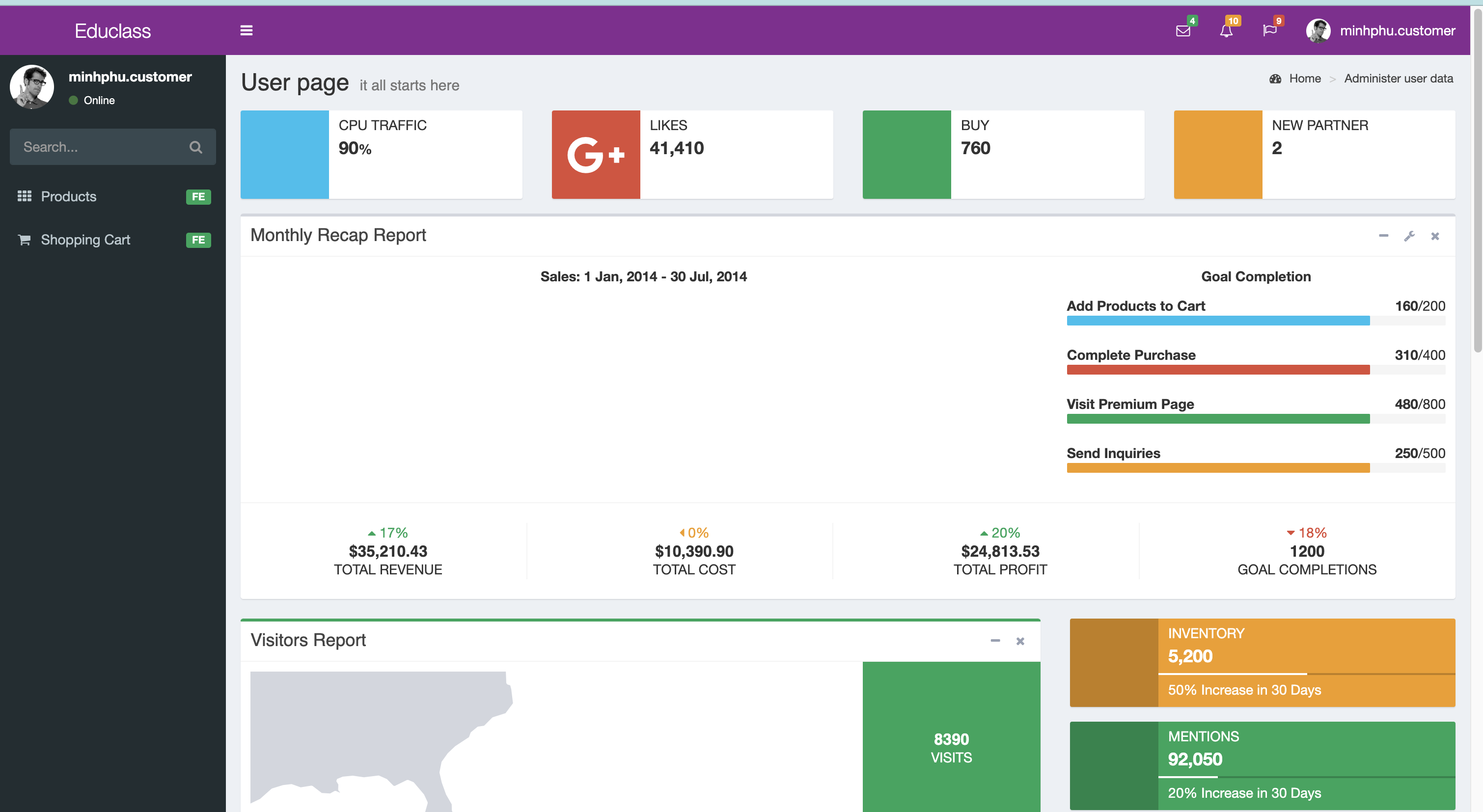
1. Allow the consumers to register in the portal, show a thank you page & send a registration confirmation email.



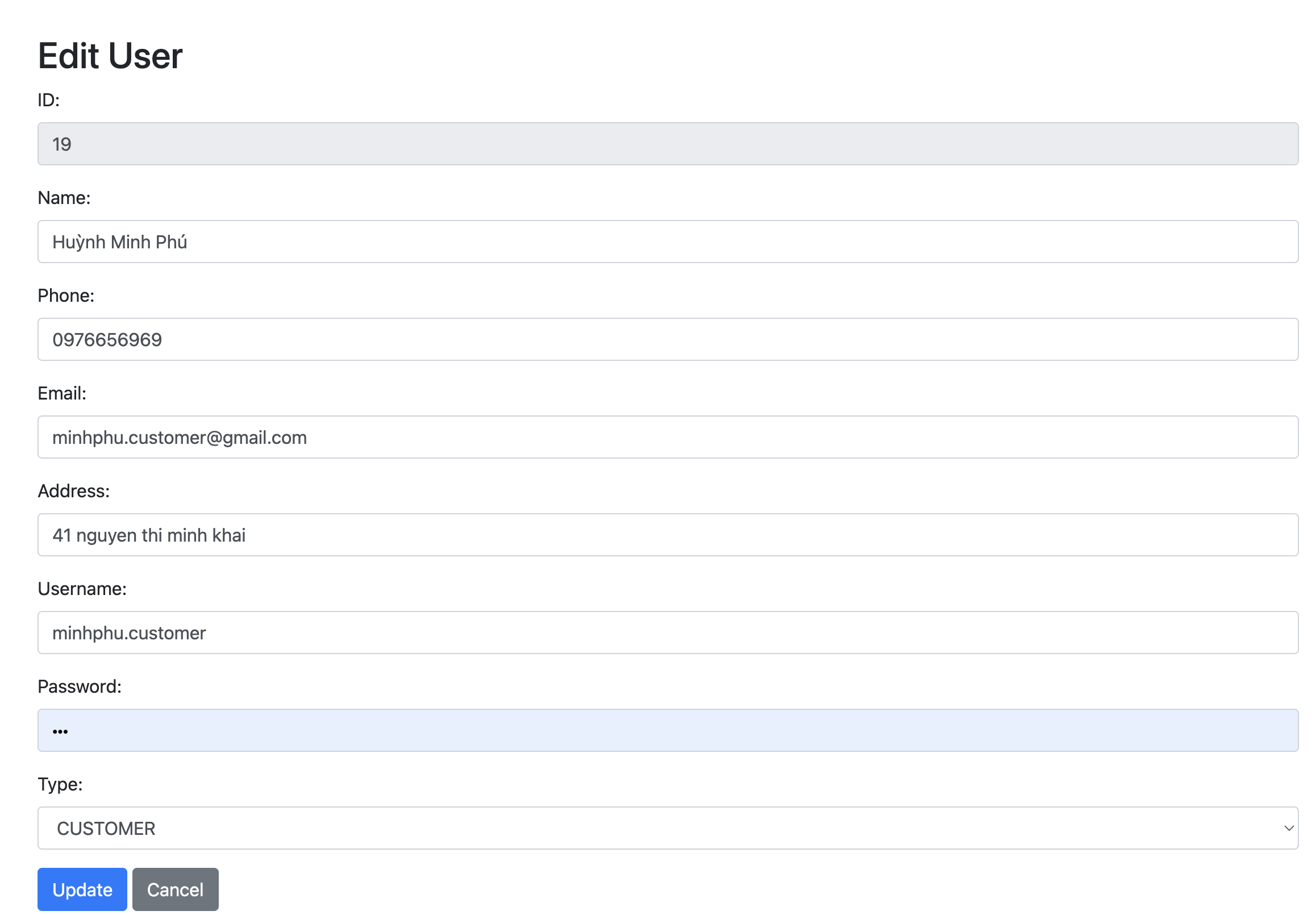


1. Search & Find Products after login and view their profile.
2. Provide Login Page

****

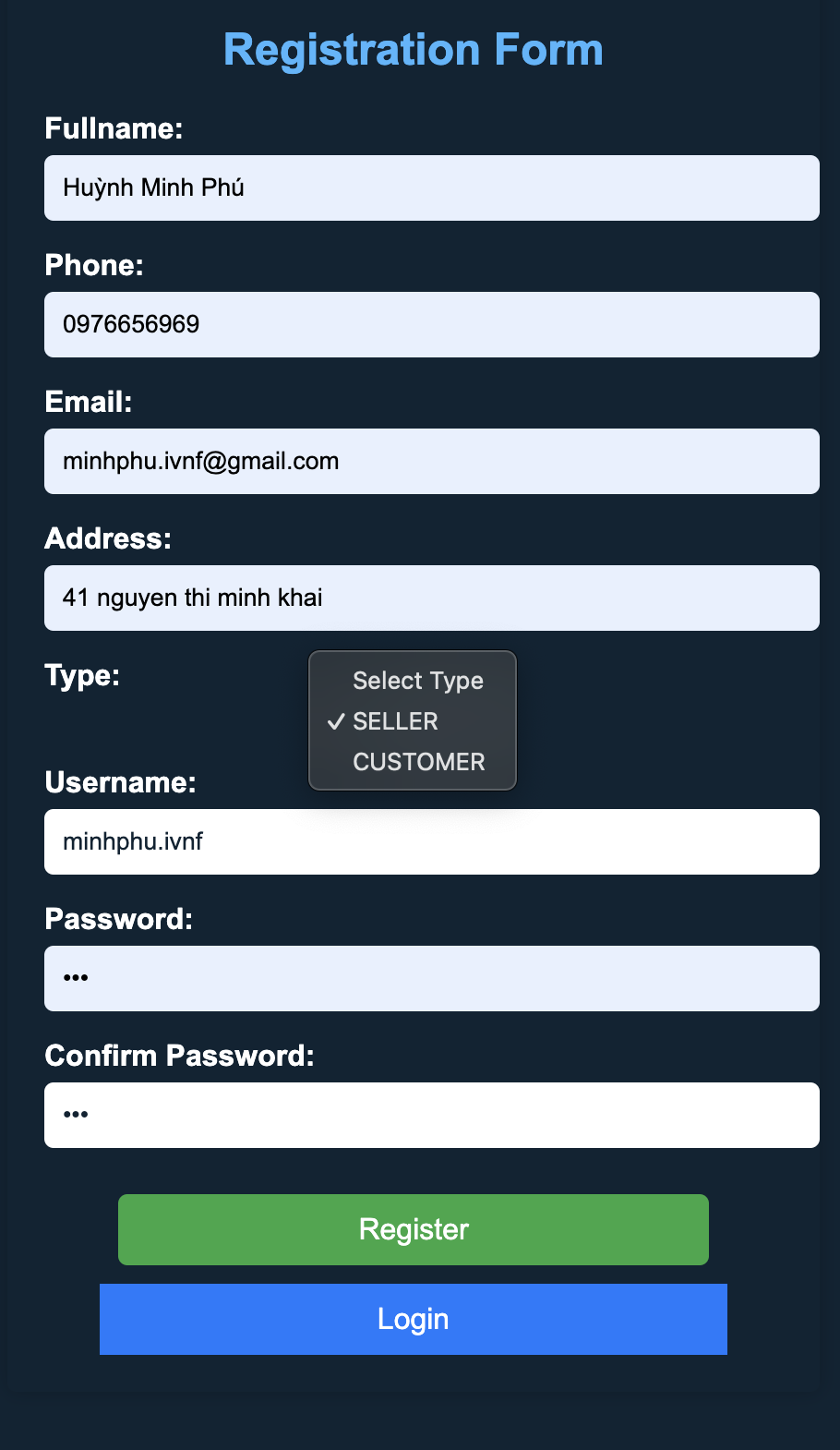
****

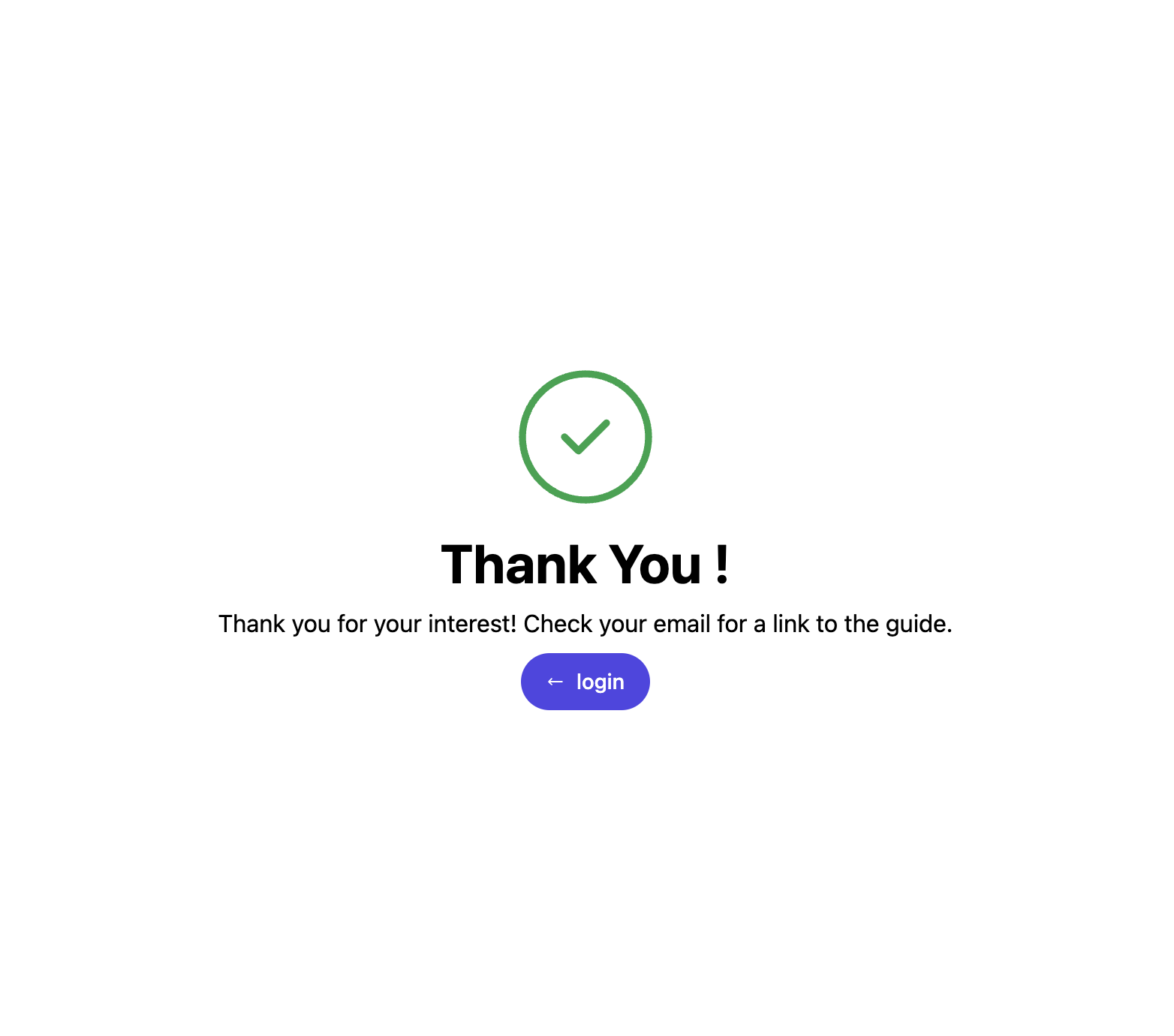
1. Provide password retrieval functionality.
2. Update their Profile after logging in.



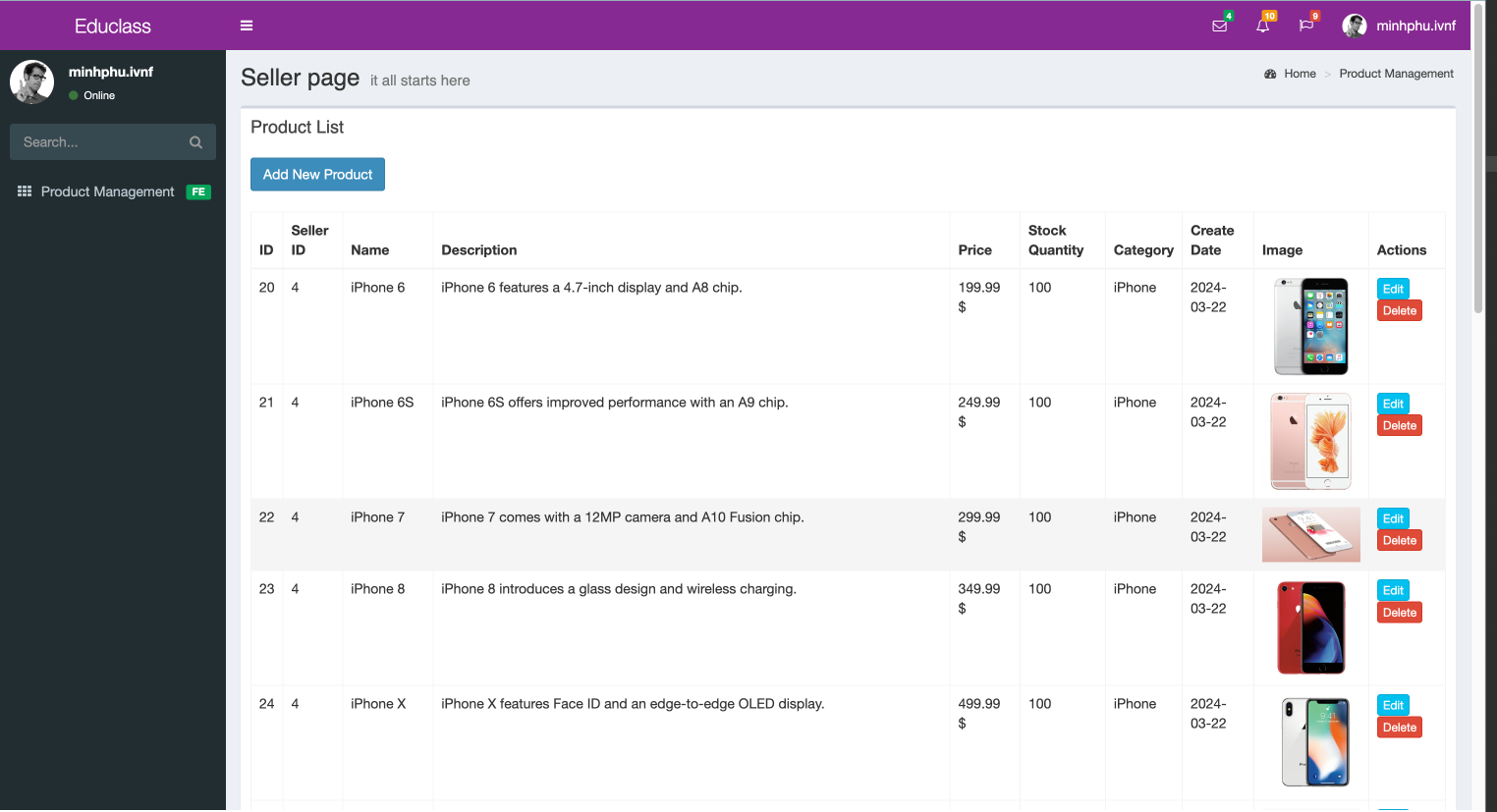
*Sellers* should be able to perform following functions in the portal

1. Allow the sellers to register in the portal, show a thank you page & send a registration confirmation email.

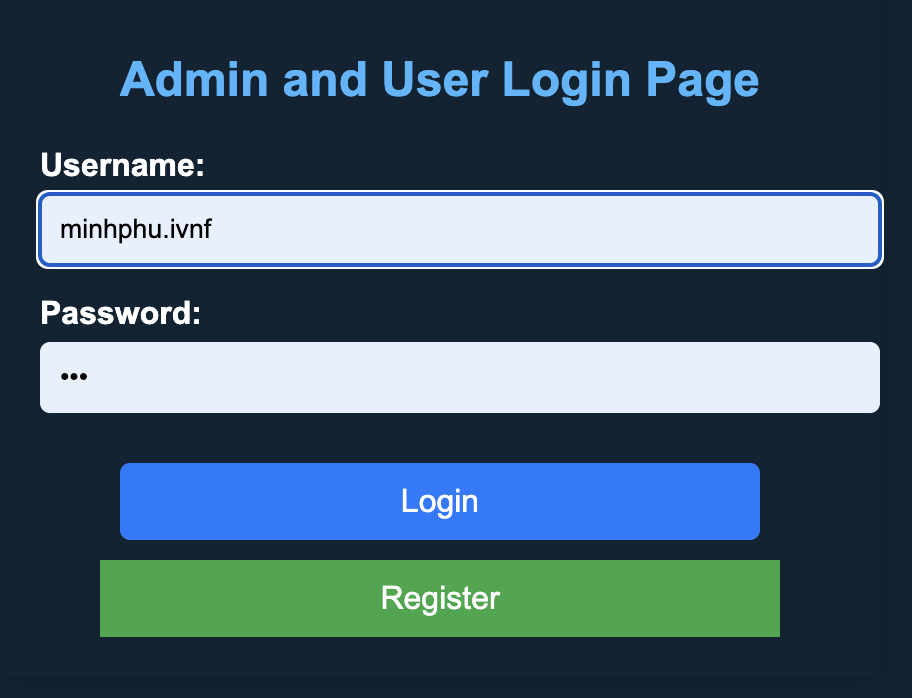




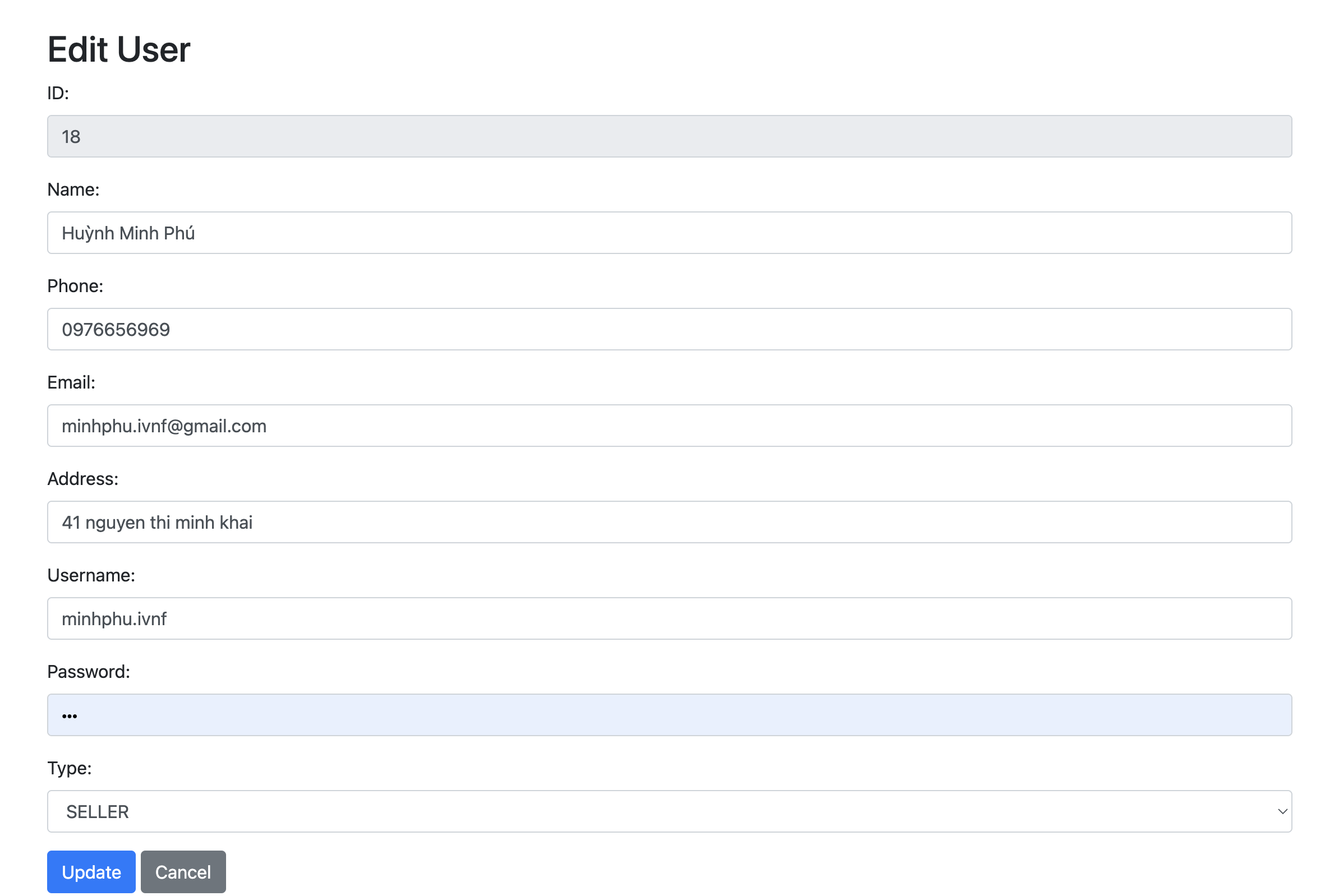
1. Maintain the product catalogue after login and view their profile.



1. Provide Login Page



1. Provide password retrieval functionality.
2. Update their Profile after logging in.



The site does not need a backend integration (Back end integration will be done in Mini Project 3), but all other display views, java script validations, url mappings etc.

* 1. Provide screen capture of HTML Pages with Spring forms.

Provide sources of Models

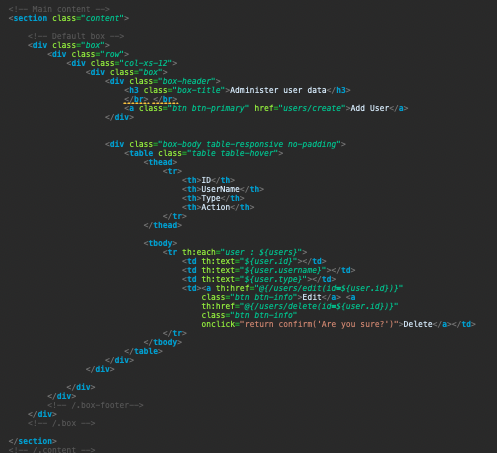
* Login Page



* Register Page



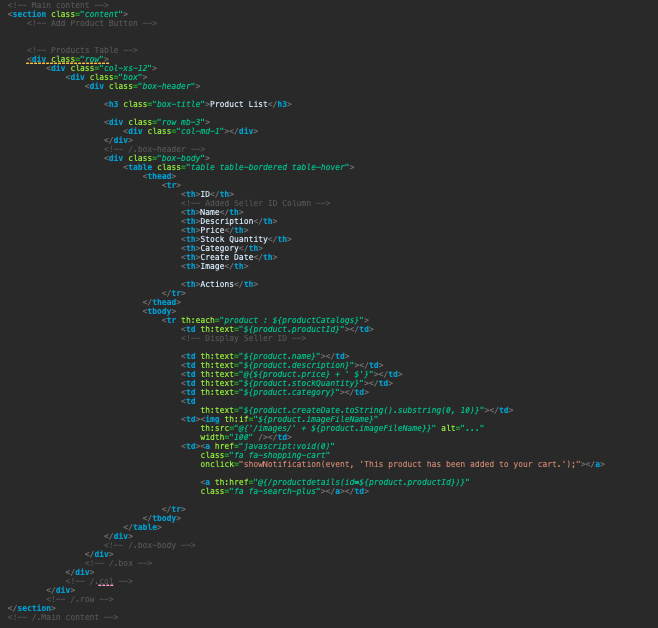
* User page



* Product page



* ShoppingCart Page



* 1. **Provide sources of Models**
* **User – Model**

package admin\_user.model;

import java.util.Date;

import java.util.HashSet;

import java.util.Set;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.OneToMany;

import jakarta.persistence.OneToOne;

import jakarta.persistence.PrePersist;

import jakarta.persistence.Table;

import jakarta.persistence.UniqueConstraint;

@Entity

@Table(name = "portal\_user", uniqueConstraints = @UniqueConstraint(columnNames = "username"))

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String type;

private String username;

private String password;

@Column(name = "create\_date")

private Date createDate;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "profile\_id", referencedColumnName = "id")

private Profile profile;

@OneToMany(mappedBy = "seller", fetch = FetchType.LAZY, cascade = CascadeType.ALL)

private Set<ProductCatalog> products = new HashSet<>();

public User() {

// Default constructor

}

// PrePersist annotation to automatically set the create date before persisting the user

@PrePersist

protected void onCreate() {

createDate = new Date();

}

// Standard getters and setters

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public Date getCreateDate() {

return createDate;

}

public void setCreateDate(Date createDate) {

this.createDate = createDate;

}

public Profile getProfile() {

return profile;

}

public void setProfile(Profile profile) {

this.profile = profile;

}

public Set<ProductCatalog> getProducts() {

return products;

}

public void setProducts(Set<ProductCatalog> products) {

this.products = products;

}

// Method to add a product to the user's product set

public void addProduct(ProductCatalog product) {

products.add(product);

product.setSeller(this);

}

// Method to remove a product from the user's product set

public void removeProduct(ProductCatalog product) {

products.remove(product);

product.setSeller(null);

}

}

* **Profile Model**

package admin\_user.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "profile")

public class Profile {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String phone;

private String email;

private String address;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}