

Lab 6 – Question 2

- Amazon Website -

Models:

1. User
2. Product
3. Order

User Model Validations

Items that should be validated: username, user email, password, mobile number, age.

*Username:

```
@NotEmpty(message = "Name can't be empty!")  
@Size(min= 7, max= 20, message = "Name length/number of letters must be in this range(7-20)")  
@Pattern(regexp = "[A-Za-z]+$", message = "Name must be letters only without numbers!")  
  
Private String name;
```

*User Email:

```
@NotEmpty(message = "Email can't be empty!")  
@Email(message = "Invalid email format!")  
  
Private String email;
```

*Mobile number:

```
@NotEmpty(message = "Mobile number can't be empty!")  
@Pattern(regexp = "^05\\d{8}$", message = "Mobile number must start with '05' and be 10 digits.")  
  
Private String mobileNumber;
```

*Password:

```
@NotEmpty(message = "Password can't be empty!")  
@Pattern(regexp = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#&()-[{}];',?/*~$^+=<>]).{8,20}$",  
message = "Password must contain at least one upper case, lower case, number, and special character.")  
  
Private String password;
```

Lab 6 – Question 2

***Age:**

@Positive(message = "Age must be a number only!")

@Min(18)

Private int age;

Potential consequences:

-If user entered invalid email, she/he can't create an account.

Product Model Validations

Items that should be validated: Product name, ID, price, description, quantity.

***Product name:**

@NotEmpty(message = "Product name can't be empty!")

@Size(min= 10, message="Product name length/number of letters must be at least 10 letters.")

Private String productName;

***Product ID:**

@NotEmpty(message = "Product ID can't be empty!")

@Size(min= 2, max= 4, message="Product ID length must be in this range(2-4).")

Private String ID;

***Product price:**

@Positive(message = "Product price must be a positive number!")

Private double price;

***Product description:**

@Size(min= 20, max= 600, message="Product description length/number of letters must be in this range(20-600).")

Private String description;

Lab 6 – Question 2

***Product quantity:**

@Positive(message= "Product quantity can't be negative number!")

Private int quantity;

Potential Consequences:

-If the product ID (unique value) allow to be empty, the same product maybe duplicated.

Order Model Validations

Items that should be validated: Order ID, user id, order price, status, shipping address.

***Order ID:**

@NotEmpty(message = "Order number can't be empty!")

@Pattern(regexp= "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).*\$", message= "Order ID must contain numbers, lower case, and upper case letters.")

@Size(min = 8, max = 8, message = "Order ID must be exactly 8 characters.")

Private String ID;

***User email:**

@NotEmpty(message = "User email can't be empty!")

Private String userEmail;

***Order price:**

@Positive(message = "Order total price must be positive number only!")

Private double order_price;

***Order status:**

@NotEmpty(message = "Order status can't be empty!")

@Pattern(regexp = "^(Cancelled|Pending|Shipped|Delivered)\$", message= "Order status must be \"Cancelled\" or \" Pending\" or \" Shipped\", or \" Delivered\" only!")

Private String status;

Lab 6 – Question 2

*Shipping address:

@NotEmpty(message = "Shipping address can't be empty!")

@Size(max= 100, message="Shipping address can't be more than 100 characters!")

Private String shippingAddress;

Lab 6 – Question 2

Lab 6 – Question 2