# **C# Web Development Basics**

# Regular Exam - CHUSHKA

CHUSHKA (Central Hierarchically-Universal Sales Host Kickstarter Application) is a universal web application for selling products, like a web shop. You have been tasked to implement this application by the Codex Input / Output organization. There are several requirements you must follow in the implementation.

# **Technological Requirements**

- Use the Web Server
- Use the MVC Framework
- Use Entity Framework

The Technological Requirements are ABSOLUTE. If you do not follow them, you will NOT be scored for other Requirements.

Now that you know the **Technological Requirements**, let us see what are the **Functional Requirements**.

# **Database Requirements**

The **Database** of the **CHUSHKA** application needs to support **3 entities**:

#### User

- Has an Id (of course) an integer
- Has an **Username**
- Has a **Password**
- Has a Full Name
- Has an **Email**
- Has an **Role** can be one of the following values ("**User**", "**Admin**")

#### **Product**

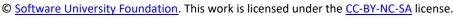
- Has a Name
- Has a **Price** think of the right data type here
- Has a **Description**
- Has a Type can be one of the following values ("Food", "Domestic", "Health", "Cosmetic", "Other")

#### Order

- Has an **Id** a **String (GUID)**
- Has a Product a Product
- Has a **Client** an **User**
- Has an Ordered On a DateTime

Implement the entities with the correct datatypes.



















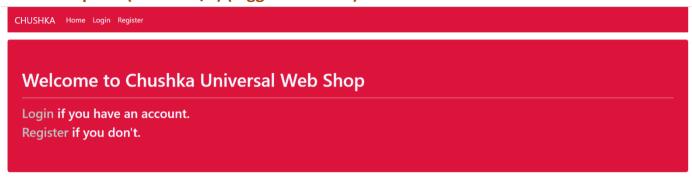


# **Template Requirements**

## **Guest Templates**

These are the templates and functionalities, accessible by Guests (logged out users).

Index Template (route = "/") (logged out user)



© CopyRight Sanity Web Design Studios 2018. All rights reserved.

## Login Template (route = "/users/login") (logged out user)



© CopyRight Sanity Web Design Studios 2018. All rights reserved.











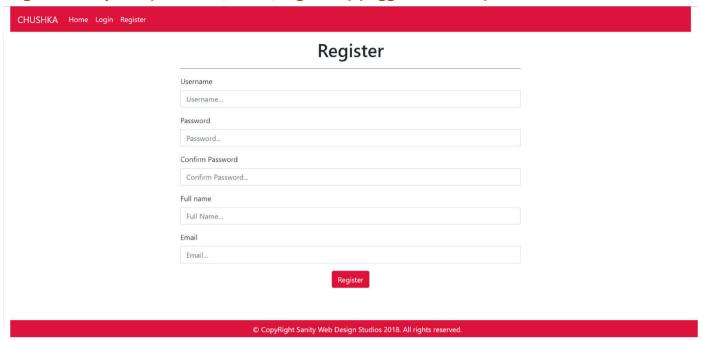








### Register Template (route = "/users/register") (logged out user)

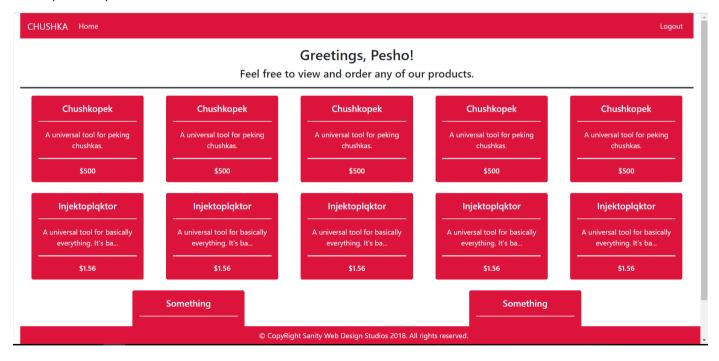


## **User Templates**

These are the **templates** and **functionalities**, accessible by **Users** (**logged in** users with **Role** - **User**).

#### LoggedIn Index Template (route="/") (logged in user)

NOTE: Products on this page have their description substringed to the 50th symbol, and then accompanied by 3 dots "..."











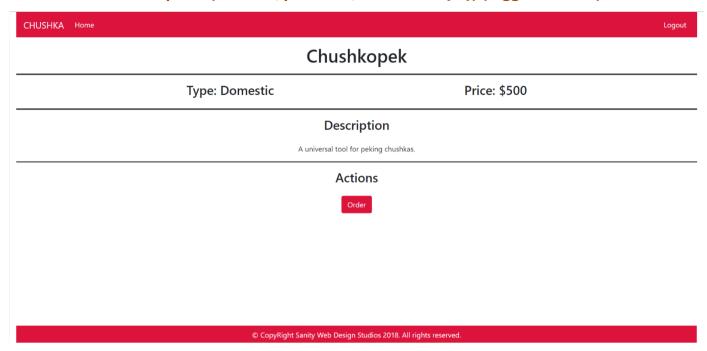








### Product Details Template (route="/products/details?id={id}) (logged in user)

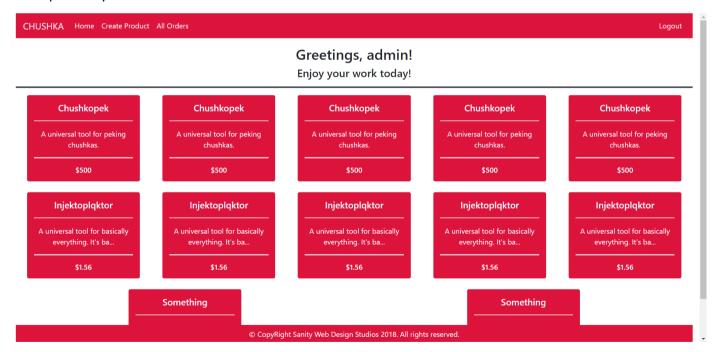


### **Admin Templates**

These are the templates and functionalities, accessible by Admins (logged in users with Role - Admin).

#### Admin Index Template (route="/") (logged in admin)

NOTE: Products on this page have their description substringed to the 50th symbol, and then accompanied by 3 dots "..."













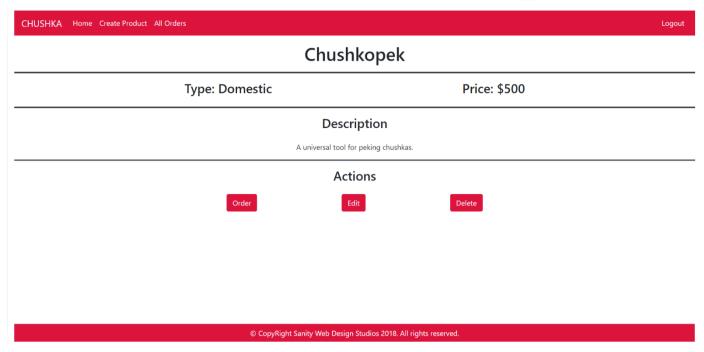




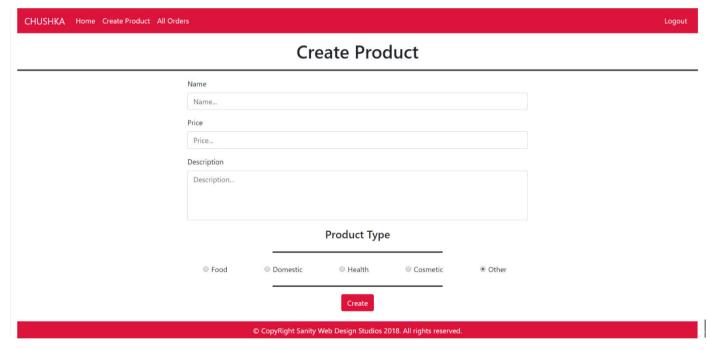




# Prod. Details Admin Temp. (route="/products/details?id={id}") (logged in admin)



# **Product Create Template (route="/products/create") (logged in admin)**















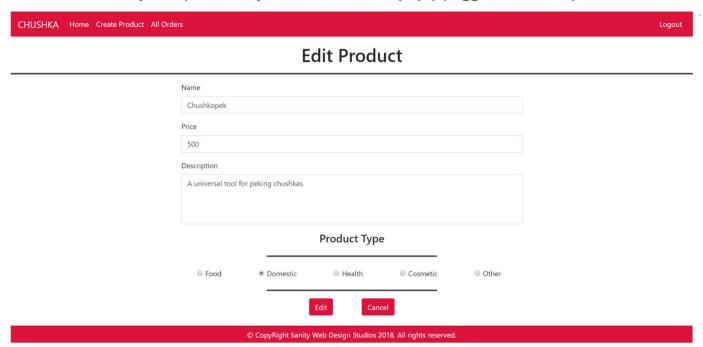




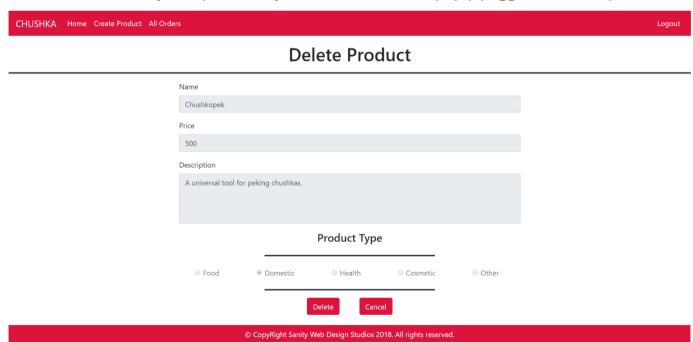




# Product Edit Template (route="/products/edit?id={id}") (logged in admin)



### Product Delete Template (route="/products/delete?id={id}") (logged in admin)















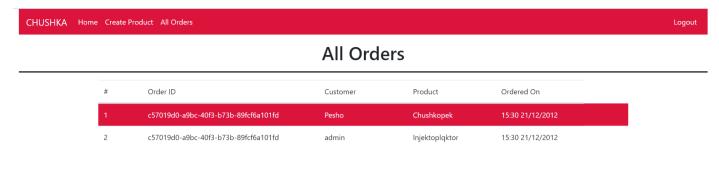








#### All Orders Template (route="/orders/all") (logged in admin)



© CopyRight Sanity Web Design Studios 2018. All rights reserved.

Some of the templates have been given to you in the application skeleton, but the others will be for you to implement, so make sure you implement them correctly. You can use the given ones as helpers.

**NOTE**: The templates should look **EXACTLY** as shown above.

**NOTE**: The templates do **NOT require additional CSS**. Only **bootstrap** and the **given css** are enough.

# **Functional Requirements**

The Functionality Requirements describe the functionality that the Application must support.

The application should provide Guest (not logged in) users with the functionality to login, register and view the **Index** page.

The application should provide Users (logged in) with the functionality to logout, and view all products, and order a Product (clicking on the [Order] button on the Product Details page).

The application should provide Admins (logged in) with the functionality to logout, and view all products, order a Product, create a Product, edit a Product, delete a Product, and view all Orders.

#### **Users**

When you register a new **User**, it should be assigned with a **role** – **User**.

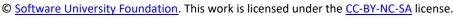
NOTE: To test your Admin functionality, make one user an Admin through the database, or make the first registered user - an Admin.

#### **Products**

Upon clicking on the rectangular object holding the data about a certain Product on the User Index and Admin Index pages, you should be redirected to its Details page.

Upon ordering a Product, an Order should be created and persisted.





















The application should store its data into a MSSQL database, using Entity Framework Core.

# **Security Requirements**

The **Security Requirements** are mainly access requirements. Configurations about which users can access specific functionalities and pages.

- **Guest** (not logged in) users can access **Index** page and functionality.
- **Guest** (not logged in) users can access **Login** page and functionality.
- **Guest** (not logged in) users can access **Register** page and functionality.
- Users (logged in) can access User LoggedIn Index page and functionality.
- **Users** (logged in) can access **User Product Details** page and functionality.
- Users (logged in) can access the Order Product functionality.
- Users (logged in) can access Logout functionality.
- Admins (logged in) can access every functionality a normal logged in User can.
- Admins (logged in) can access Admin LoggedIn Index page and functionality.
- Admins (logged in) can access the Admin Product Details page and functionality.
- **Admins** (logged in) can access the **Product Create** page and functionality.
- **Admins** (logged in) can access the **Product Edit** page and functionality.
- **Admins** (logged in) can access the **Product Delete** page and functionality.
- **Admins** (logged in) can access the **All Orders** page and functionality.

# **Scoring**

This section describes how the scoring of the Exam Preparation will be made.

Database Requirements – 15 points.

Template Requirements – 25 points.

Functional Requirements – 50 points.

Security Requirements – 10 points.



















