# Exercise: Web Server – Databases

Problems for exercises and homework for the [“C# Web Basics” course @ SoftUni](https://softuni.bg/courses/csharp-web-development-basics). Yoy can submit your solution in the course web page.

Back to our By The Cake web site. We will connect our web site with a database so we can keep track of our data easily.

## Create a Database Connection

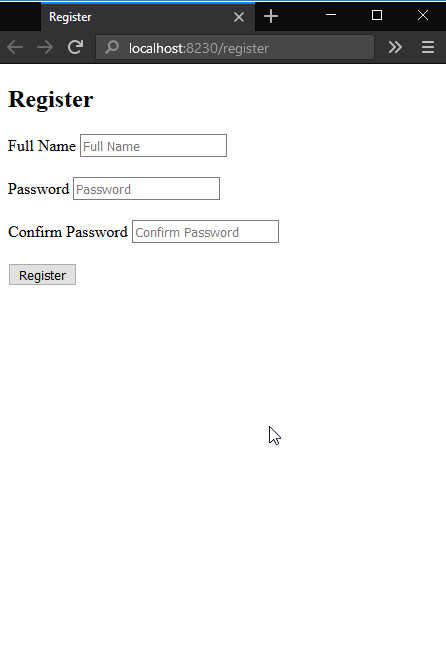
Creating the database connection is pretty straight forward. We should add a project to keep our **context class** and a project to keep our **models**. If you iplement the **Repository pattern** and/or the **Unit of Work pattern** (**and this is highly recommended**), make sure they are in their own projects.

Start with creating the models. We will have **users**, **orders** and **products**.

The **user** has **name**, **username**, **password**, **date of registration** and a **collection of orders**. The **order** has **date of creation** and a **collection of products**. The **product** has **name**, **price** and **image URL**. **Each product** can be part of **many orders**.

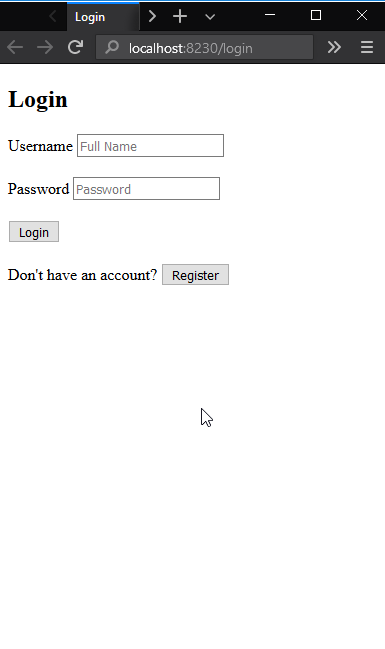
## Register Users

Create a Register page with a simple form. The form should ask for a username and a password. When a user is registered successfully, he/she should be taken to the home page.



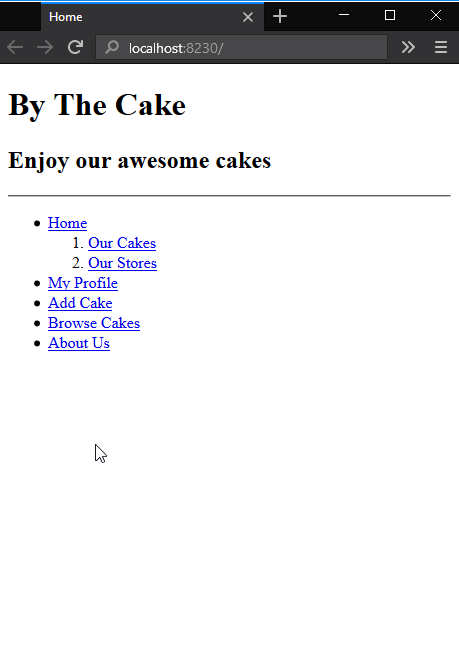
## Login Users

When the user try to log in you should check if the user exists in you database. If there is no such user, you should navigate to the Register page and ask the user to register.

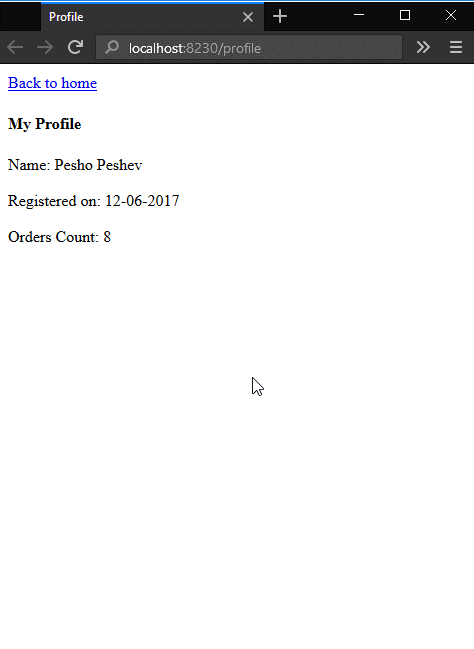
Add a link to the Register page and style it as a button.

## User Profile Page

Add link **My Profile** to the **Home** page menu.



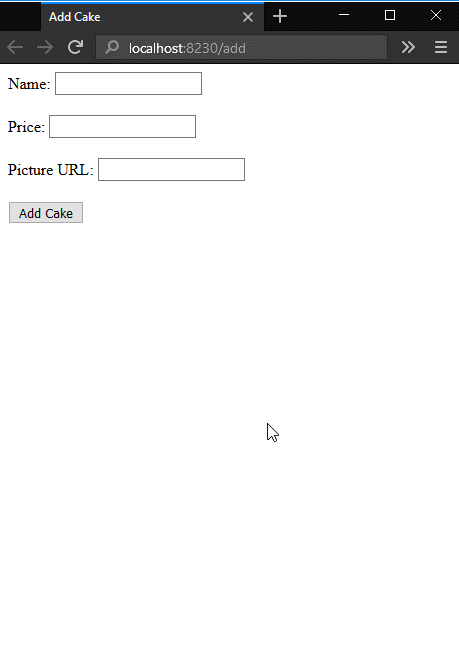
The link should navigate to the page **Profile** where you can see details about the user. The page should display the **name**, the **username**, the **date registered** and the **count of orders** of the logged in user.



## Add Cakes to the Database

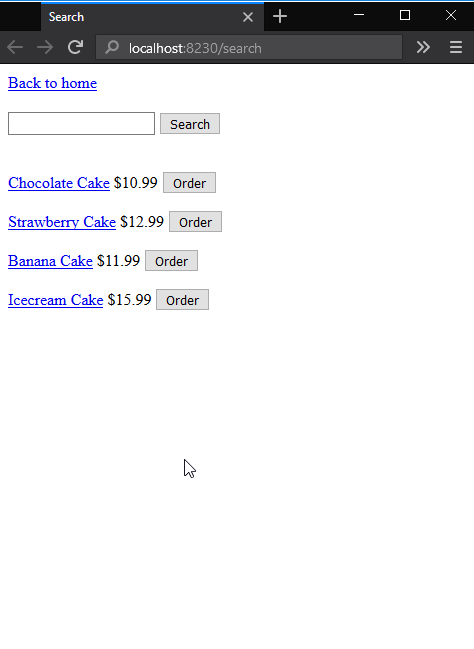
On the **Add Cake** page the user can add cakes. You should store those cakes in the database. Extend the form and add a field for the URL of the picture of the cake.

Fill some cakes so we can use them to test the application later.



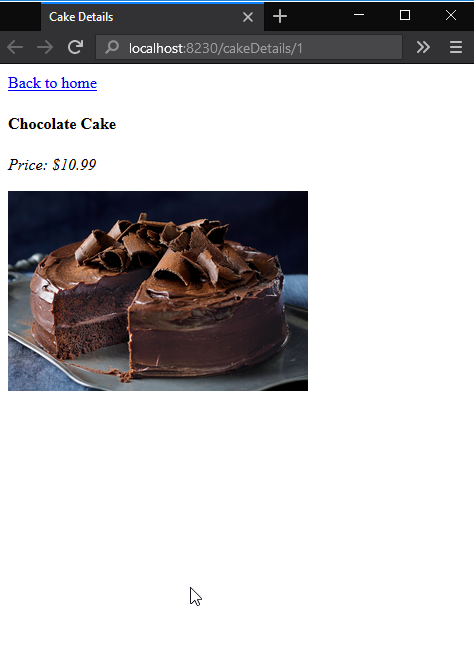
## The search Page

Now we are going to search all the cakes from our database. When the browser loads the page, there should be listed all cakes in the database. If there is no such cake, show a "Cake Not Found" message. The names of the cakes should be links that navigate to a page with details about the cake with a URL "**localhost:{port}/cakeDetails/{cake id}**".



## Show Cake Details

Create the page **CakeDetails** and display the name, the price and the picture of the cake.

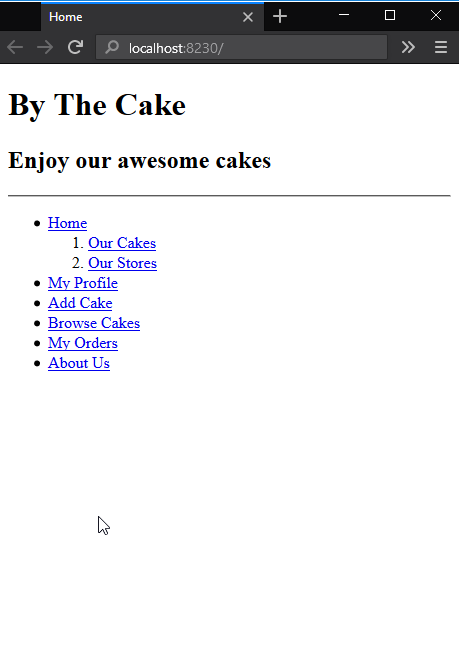


## Make an Order

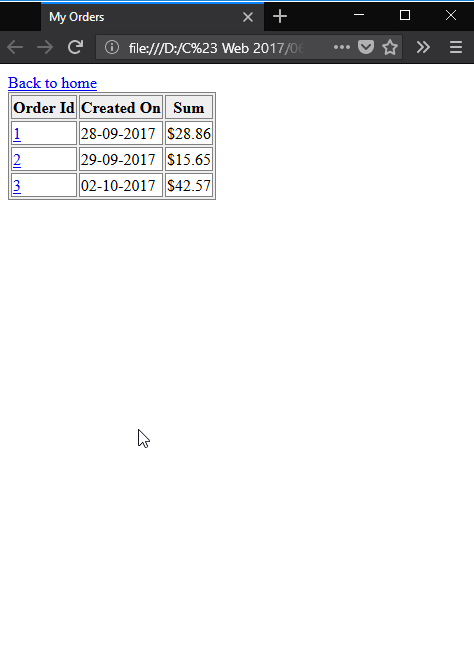
Users can order cakes, collect products in their cart and finish the order from the page Cart. Now when a user submits an order, you should keep that order in the database.

## List Orders

Add link My Orders to the home page menu.



The link should navigate to the **Orders** page ("**localhost:{port}/orders**"). That page should show a **table** with **all orders** made by the user. The orders should be ordered by date created in descending order.



The **id** of the order should be a **link** that navigates to a page with details for the given order. The path to that page is "**localhost:{port}/orderDetails/{order id}**".

## Show Order Details

The **OrderDetails** page should visualize details about the order (what a surprise!) with the given **id**. The user should see when the order is created, a list with **all products** and the **total sum** of the order.

The name of the product is a link that navigates to a page with details aboult the given cake – "**localhost:{s}/cakeDetails/{cake id}**".