Inredning Online

Masoomeh Ghasemi

2020.12.27

Structure

I started off with creating an ASP:NET core web application with an empty template. I have mainly used the MVC pattern, except for one case (creating the pool or projects which is similar to a shopping cart) for which I have used Razor pages. The main structure of the app is made of the following components:

- Models:
    - Product: for defining the product model. I have included fields for Category and Supplier directly in this class. On a second reflection, I believe it was better to use separate models for the categories and suppliers. In fact, I started off with this scenario, but due to my limited knowledge of ASP.NET core, I soon got into the case with breaking circular reference error which I did not manage to resolve. In a real case project, I would use separate models.
    - ProductRepository / IProductRepository: for interacting with the table of products in the database.
    - Project: for defining the project model. this class is similar to ShoppingCart class in Bethany's Pie Shop project.
    - ProjectDetails: This class is for defining the data model of the projects which will be saved to the database.
    - ProjectDetailsRepository / IProjectDetailsRepository: for interacting with the table of ProjectDetails
    - ProjectModel: this is counterpart of the Order class in Bethany's Pie Shop project. This is the model part of the Razor page that I have used.
    - SessionProject: inherits from Project and provides a service session for each project.
- Controllers
    - HomeController
    - ProductCotroller: ProductRepository is injected into this controller in order to list the products.
    - PrjectDetailsController: for submitting the projects and saving to the database.
    - UserPanelController: to perform the requested tasks, e.g. creating a project, listing existing projects and editing them.
    - AdminPanelController: contains the action method for the admin (Ingrid). If the View was the same for the users and for Ingrid, I could have used "UserPanelController" for both of normal user and Ingrid. But since the Views are supposed to be different according to the requirement list (Ingrid does not need Details/Edit buttons), I have added this extra controller.
- Views: corresponding views for GET/POST requests which are handled by the controllers.

NB: I have ignored listing the ViewComponents/ViewModels/Auxilary classes.

Once the application starts, the home page with some instruction is shown and the links to the User and Admin panels are provided. After logging in, the user can create a project and submit it, can list the existing projects and edit them. Snapshots of these steps are shown in the Appendix.

Security

I added authentication/authorization using the scaffolded Identity setup and made some changes to add two roles; “User” and “Admin”. More on this will be explained in the next section. In order to make sure that only logged-in users can access the application, I used [Authorize] attribute on the controllers. For the controllers that are related to the “user”, the attribute is used as `[Authorize(Roles = "user")]` and for the admin, I added `[Authorize(Roles = "admin")]`. I also added a class for the user (ApplicationUser class) when adding the Identity feature and to this app. Thus, only logged-in users could use the application.

In order to give appropriate access to “user” and “admin, in the ProjectDetails model, field were assigned to UserEmail (also, FirstName and LastName of the user). The UserEmail property is used to identify the logged-in user and list his/her projects. The name properties are used along with the projects when listed.

In the ProjectDetailsController, I have used the dependency injection of the UserManager to add the user details to each project. The highlighted parts of the below code snippt are related to this:

```
private readonly UserManager<ApplicationUser> userManager;

public ProjectDetailsController(IProjectDetailsRepository repo,
            Project projectService,
            UserManager<ApplicationUser> userMgr)
        {
            repository = repo;
            project = projectService;
            userManager = userMgr;
        }

[HttpPost]
        public async Task<IActionResult> SubmitProject(ProjectDetails projectDetails)
        {
            if (project.ProjectItems.Count() == 0)
            {
                ModelState.AddModelError("","The project is empty!");
            }

            if (ModelState.IsValid)
            {
                var user = await userManager.GetUserAsync(HttpContext.User);
                projectDetails.FirstName = user.FirstName;
                projectDetails.LastName = user.LastName;
                projectDetails.ProjectItems = project.ProjectItems.ToArray();
                projectDetails.ProjectCreated = DateTime.Now;
                projectDetails.UserEmail = user.Email;
                repository.SaveProjectDetails(projectDetails);
                project.ClearProject();
                return RedirectToPage("/Completed",
                    new {projectDetailsId = projectDetails.ProjectDetailsId});
            }
            else
            {
                return View();
            }
        }
```

Then in the UserPanelCotroller, I could filter the project for the current user using:

```
public IActionResult ListProjects()
        {
            var currentUserEmail = this.User.Identity.Name;

            AllProjects = repository
                .GetListOfProjectForCurrentUser(currentUserEmail).ToList();

            return View(AllProjects);
        }
```

As mentioned before, because the view for user and admin are different (admin cannot edit projects), I made a separate controller view for admin. Otherwise, I could use the above action method for both the user and the admin by checking their role using “`User.IsInRole(“role”)`” and adding an `if` statement in the action method.

Users

There are two types of users in the application: “admin” and “user”. After setting up the automatic Authentication and Authorization, I made the following changes:

- I added the possibility of assigning roles to the users by adding “`.AddRoles<IdentityRole>()`” to the IdentityHostingStartup.cs
- Then in the context file related to the identity database (IdintityDbContext), I seeded two users with two roles: “admin” and “user”.using `OnModelCreating()` method. I also seeded two users, one with “admin” role and one with “user” role.
- To make sure that, there is only one admin and all the new users will have the “user” role, I added the following line in the OnPostAsync action of “Register.cshtml.cs” class:
  `await _userManager.AddToRoleAsync(user, "user");`
- By assigning the value to “Roles” for the [Authorize(Roles=”role”)] attribute before controller actions (or controller), I could give access to parts of the application to appropriate user type.

The credentials for the seeded users are:

Admin:

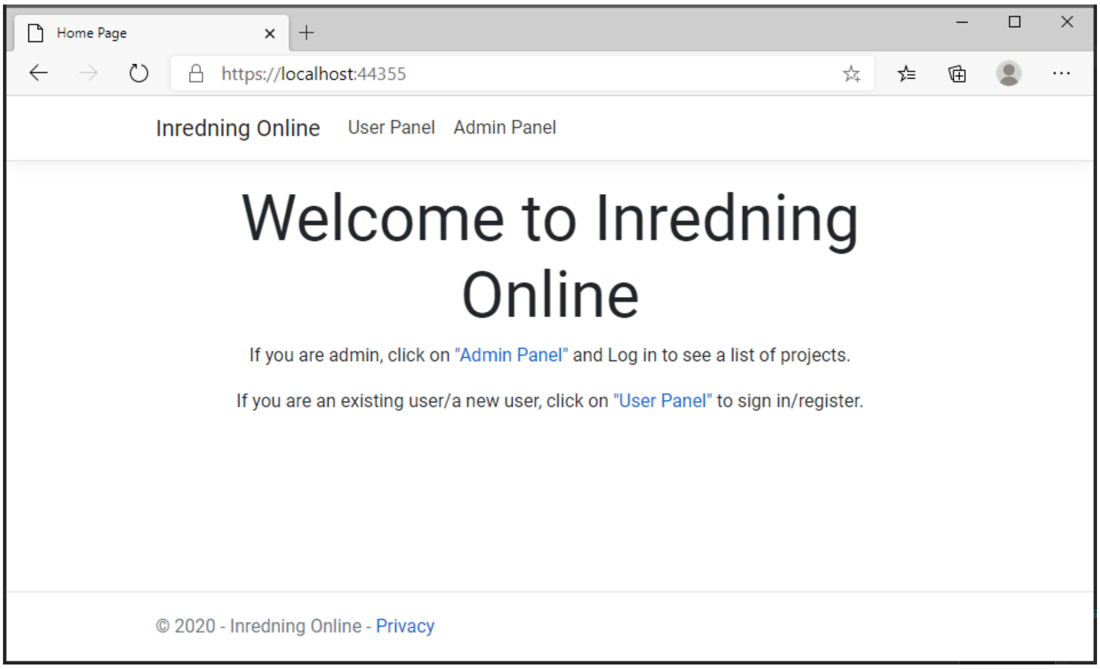
- Email: admin@admin.com
- Password: Test2020!

User:

- Email: user@user.com
- Password: Test2020!

Appendix

1. Starting the application



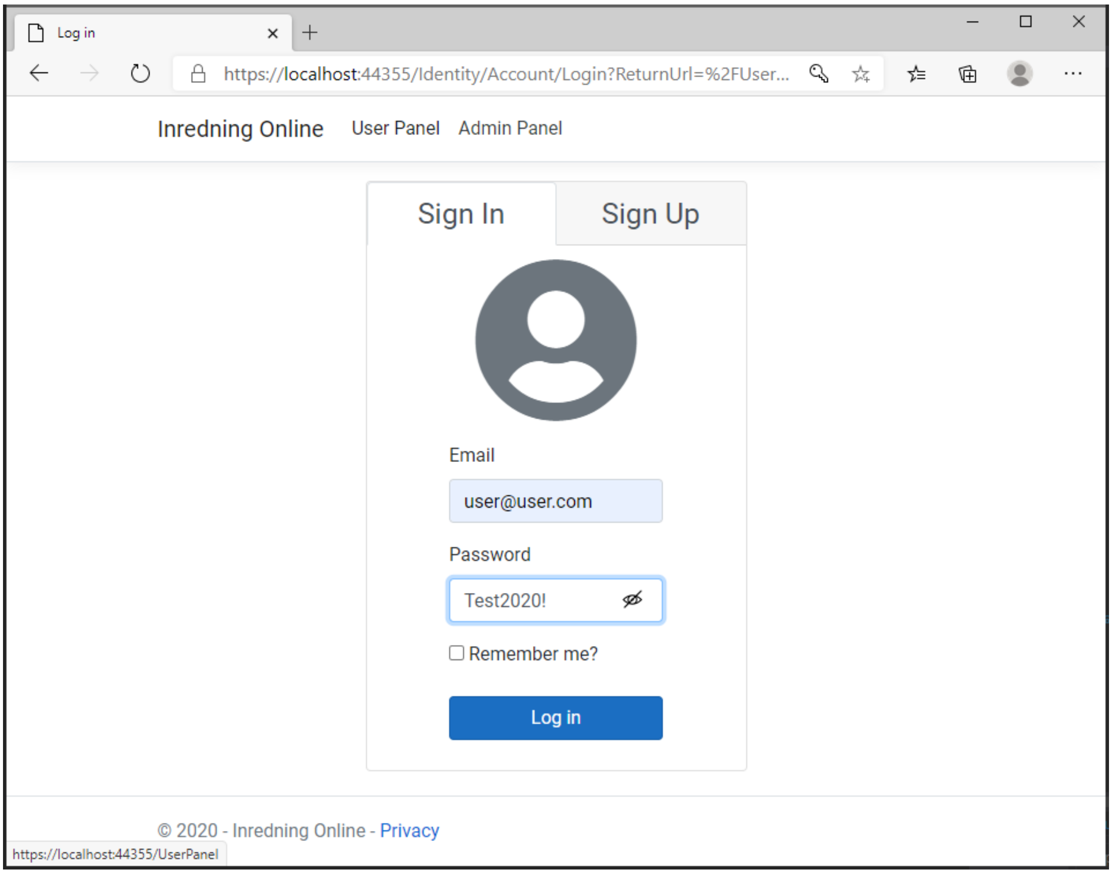
Welcome to Inredning Online

If you are admin, click on "Admin Panel" and Log in to see a list of projects.

If you are an existing user/a new user, click on "User Panel" to sign in/register.

© 2020 - Inredning Online - Privacy

2. Signing in as a regular user:



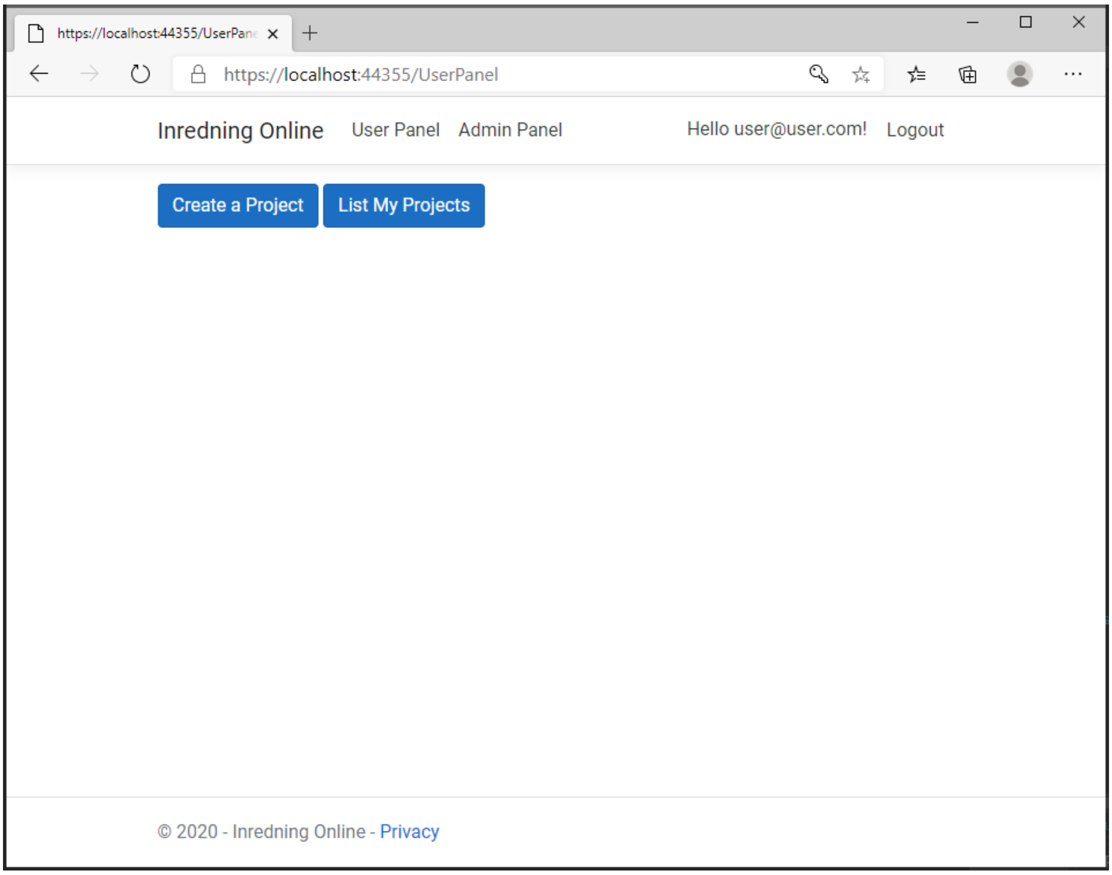
Log in
https://localhost:44355/Identity/Account/Login?ReturnUrl=%2FUser...
Inredning Online
User Panel
Admin Panel
Sign In
Sign Up
Log in
© 2020 - Inredning Online - Privacy
https://localhost:44355/UserPanel

Email: user@user.com

Password: Test2020!

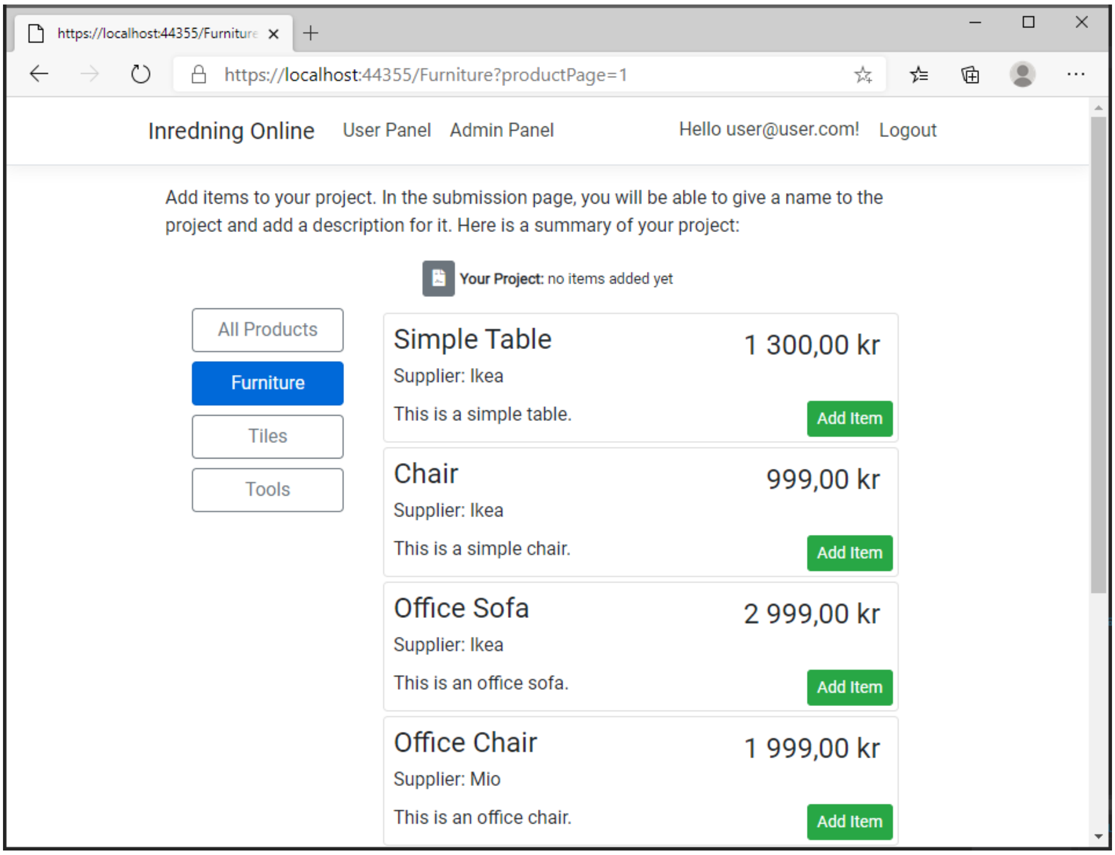
☐ Remember me?

3. A logged-in user can create a project or list the existing projects:


https://localhost:44355/UserPanel

© 2020 - Inredning Online - Privacy

4. The user starts by adding items to the project. Here, it is possible to filter for categories.


https://localhost:44355/Furniture?productPage=1

Add items to your project. In the submission page, you will be able to give a name to the project and add a description for it. Here is a summary of your project:

Your Project: no items added yet

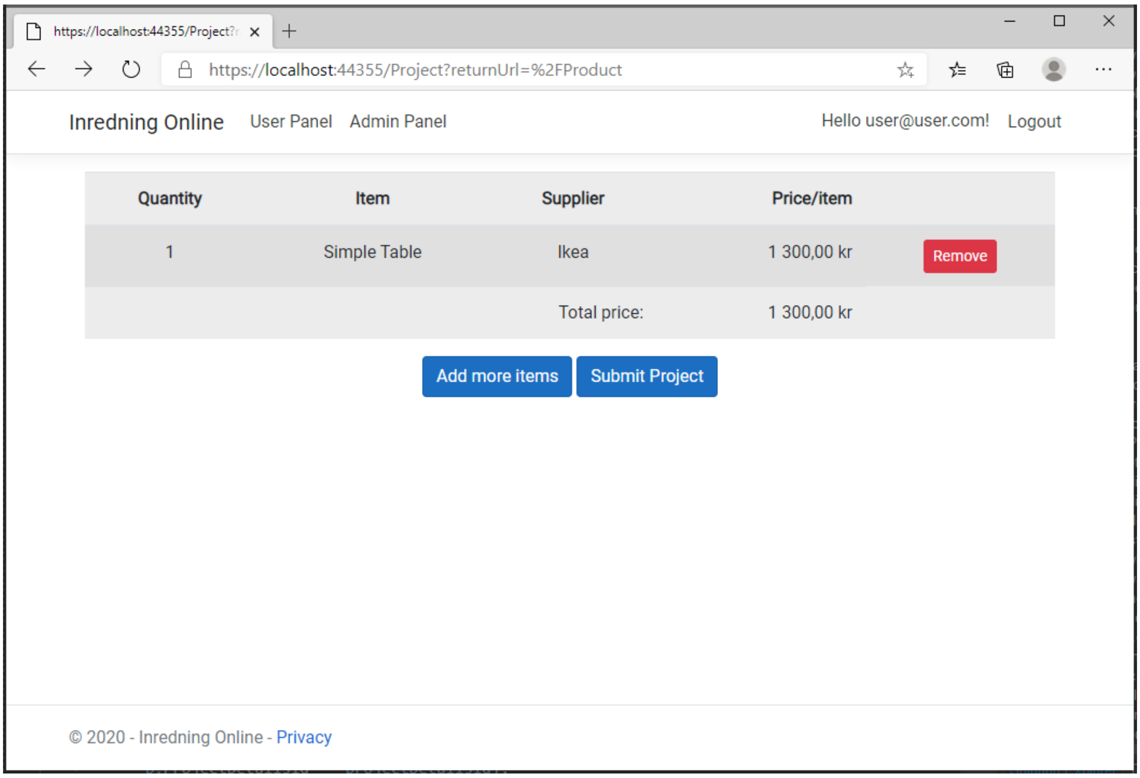
Simple Table 1 300,00 kr
Supplier: Ikea
This is a simple table.

Chair 999,00 kr
Supplier: Ikea
This is a simple chair.

Office Sofa 2 999,00 kr
Supplier: Ikea
This is an office sofa.

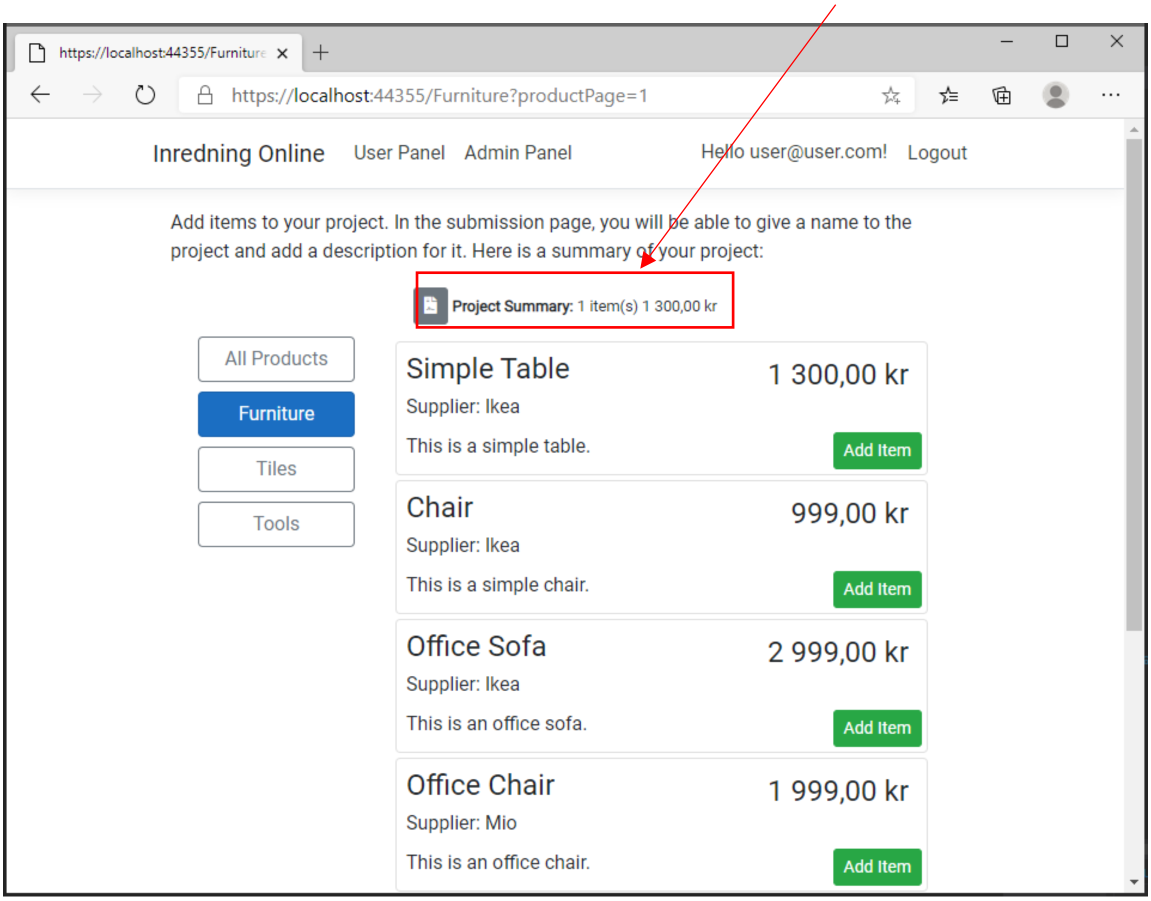
Office Chair 1 999,00 kr
Supplier: Mio
This is an office chair.

5. After adding an item, the user can add more items by clicking on the button or submit the project. The user can also remove the added item.



Quantity	Item	Supplier	Price/item	
1	Simple Table	Ikea	1 300,00 kr	Remove
		Total price:	1 300,00 kr	

6. A summary of the project is shown on top of the product catalogue.



Add items to your project. In the submission page, you will be able to give a name to the project and add a description for it. Here is a summary of your project:

Project Summary: 1 item(s) 1 300,00 kr

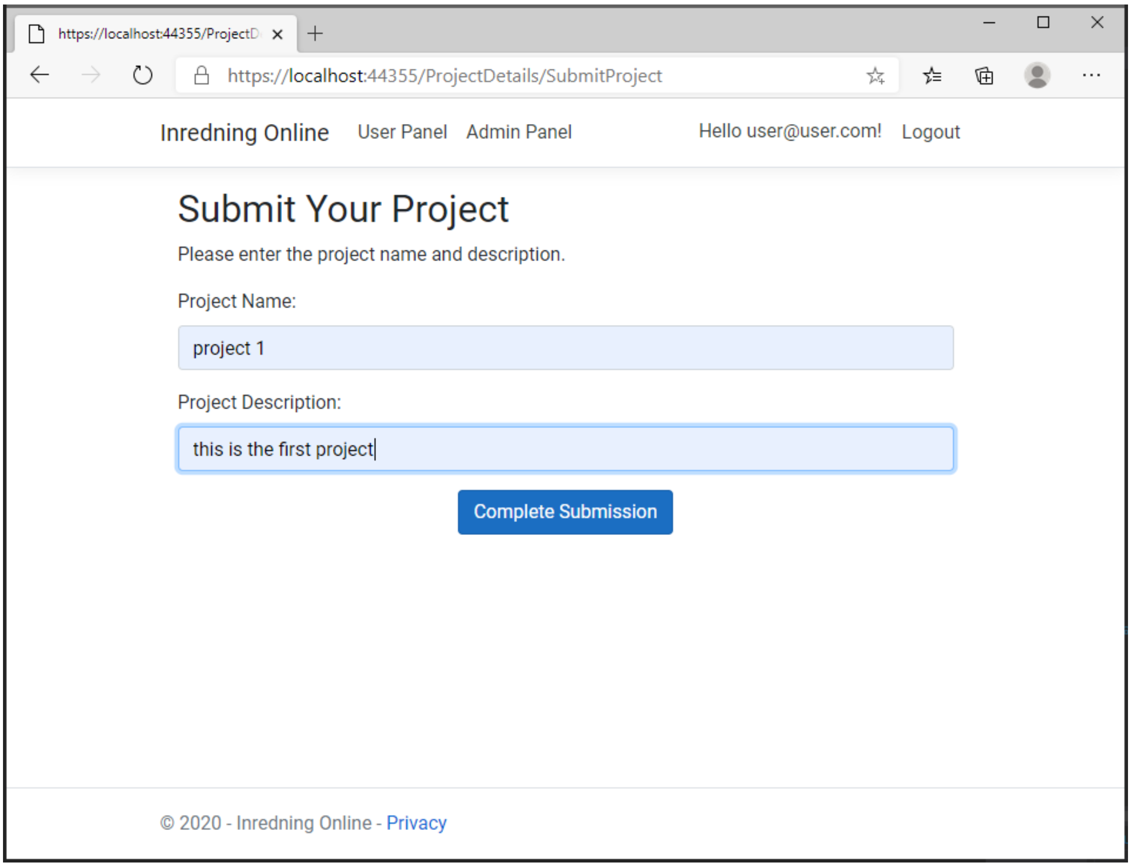
Simple Table 1 300,00 kr
Supplier: Ikea
This is a simple table.

Chair 999,00 kr
Supplier: Ikea
This is a simple chair.

Office Sofa 2 999,00 kr
Supplier: Ikea
This is an office sofa.

Office Chair 1 999,00 kr
Supplier: Mio
This is an office chair.

7. On the submission page, the user needs to enter a name a description for the project. The form validation has been used to prevent submitting a project with no item, no name and no description.



https://localhost:44355/ProjectD
https://localhost:44355/ProjectDetails/SubmitProject
Inredning Online
User Panel
Admin Panel
Hello user@user.com!
Logout

Submit Your Project

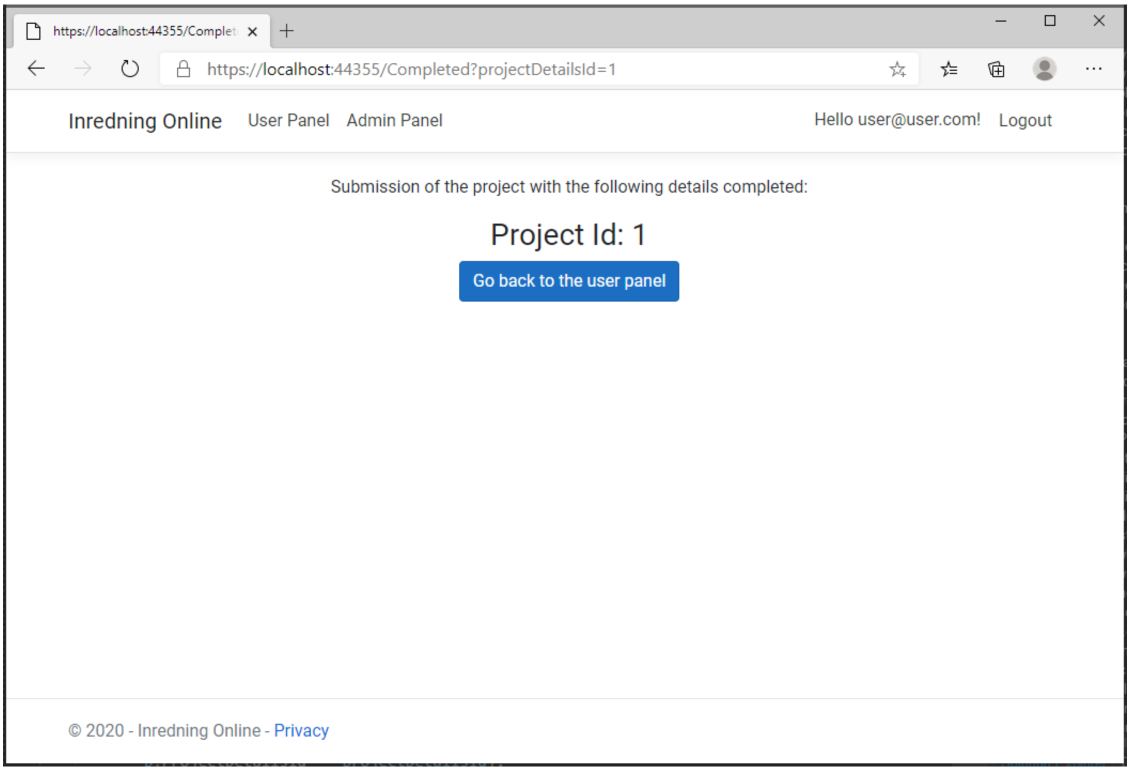
Please enter the project name and description.

Project Name: project 1

Project Description: this is the first project

Complete Submission
© 2020 - Inredning Online - Privacy

8. After submission, a message is shown.



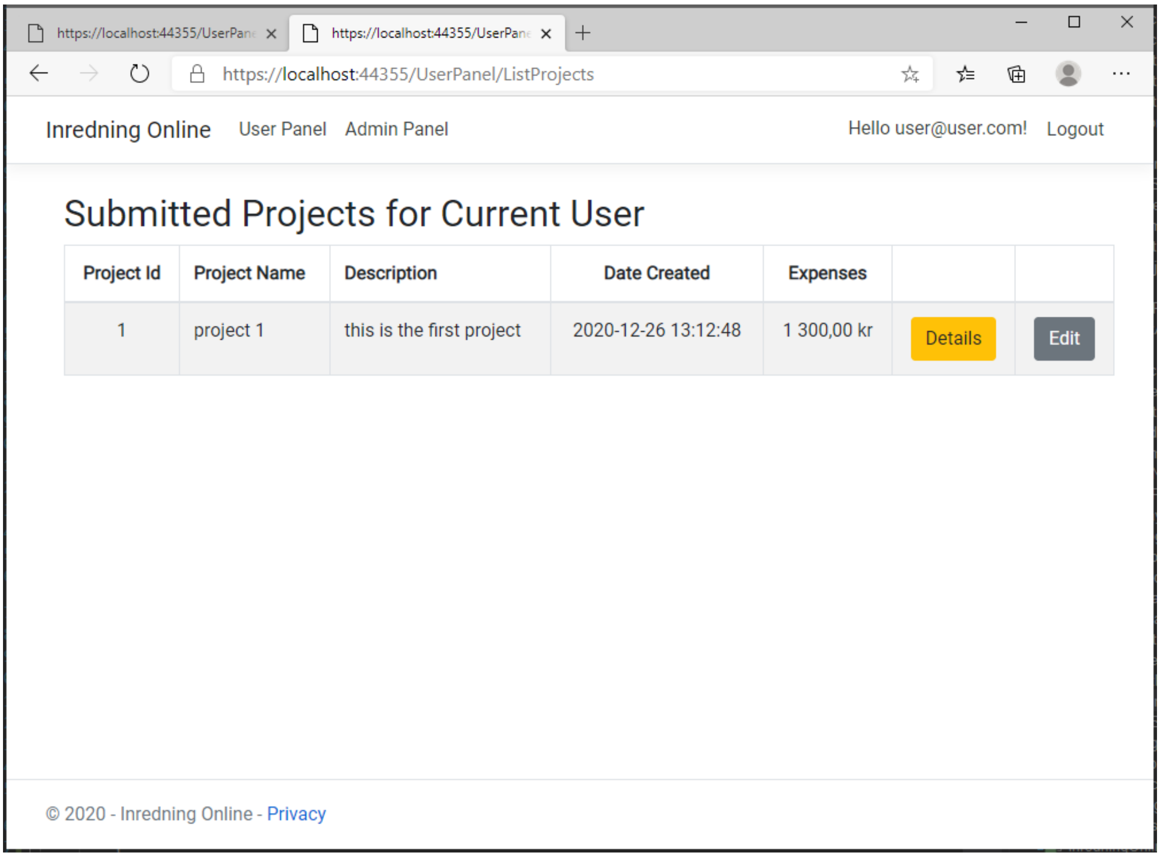
https://localhost:44355/Complet
https://localhost:44355/Completed?projectDetailsId=1
Inredning Online
User Panel
Admin Panel
Hello user@user.com!
Logout

Submission of the project with the following details completed:

Project Id: 1

Go back to the user panel
© 2020 - Inredning Online - Privacy

9. The user can list all his/her submitted project via the user panel.

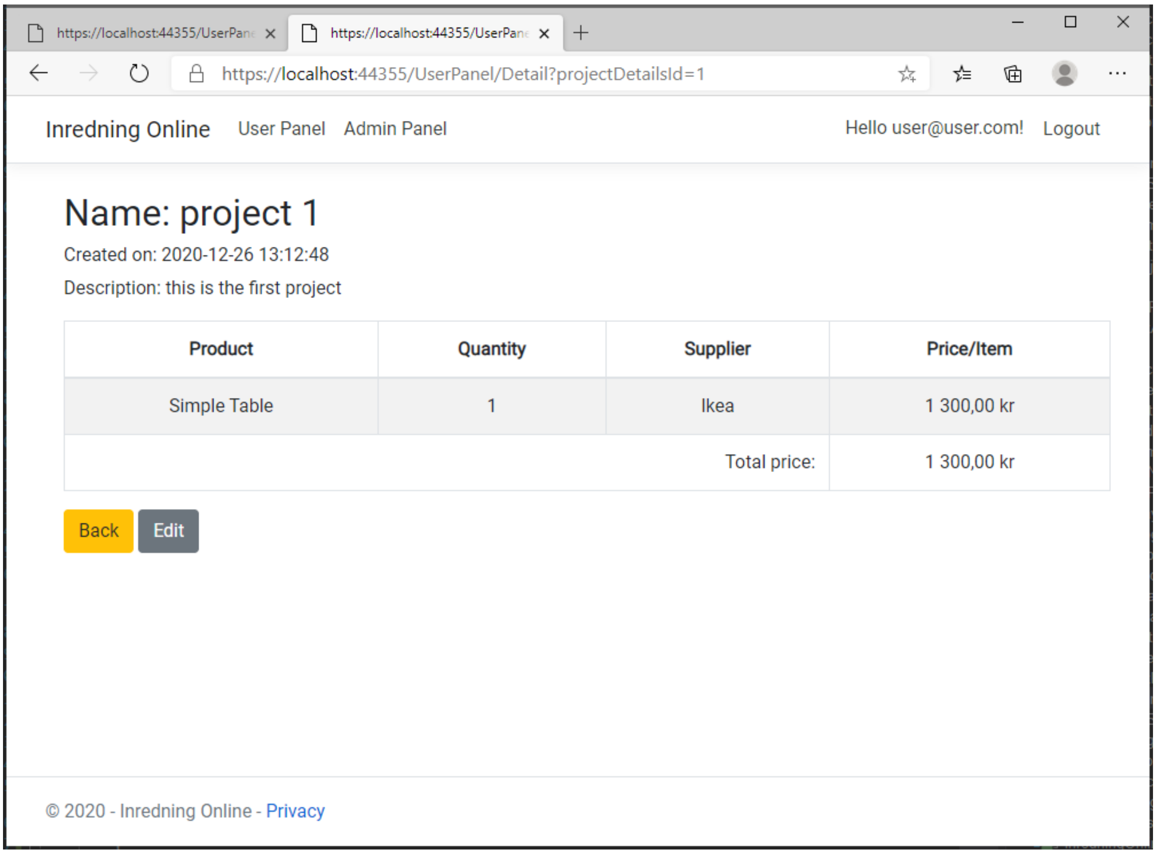


Submitted Projects for Current User

Project Id	Project Name	Description	Date Created	Expenses		
1	project 1	this is the first project	2020-12-26 13:12:48	1 300,00 kr	Details	Edit

© 2020 - Inredning Online - Privacy

10. And see the details of each project:



Name: project 1

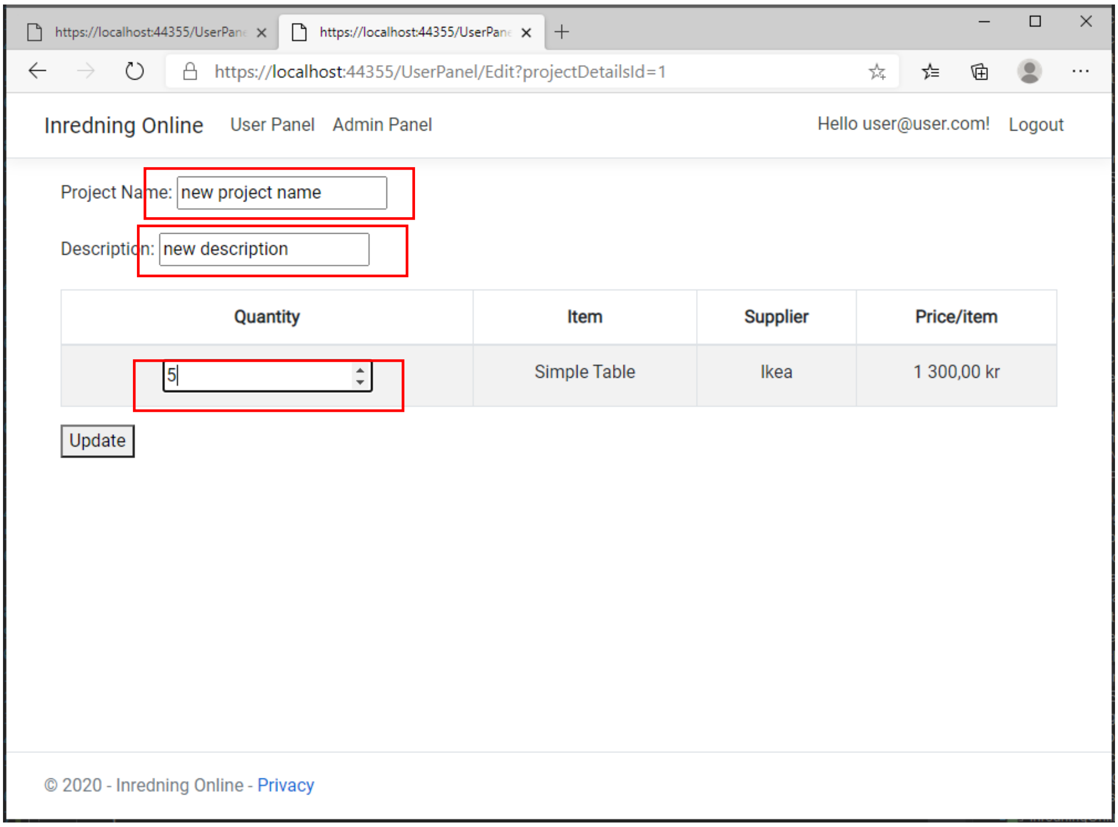
Created on: 2020-12-26 13:12:48

Description: this is the first project

Product	Quantity	Supplier	Price/Item
Simple Table	1	Ikea	1 300,00 kr
		Total price:	1 300,00 kr

© 2020 - Inredning Online - Privacy

11. The user can also edit each project (marked with red rectangles).



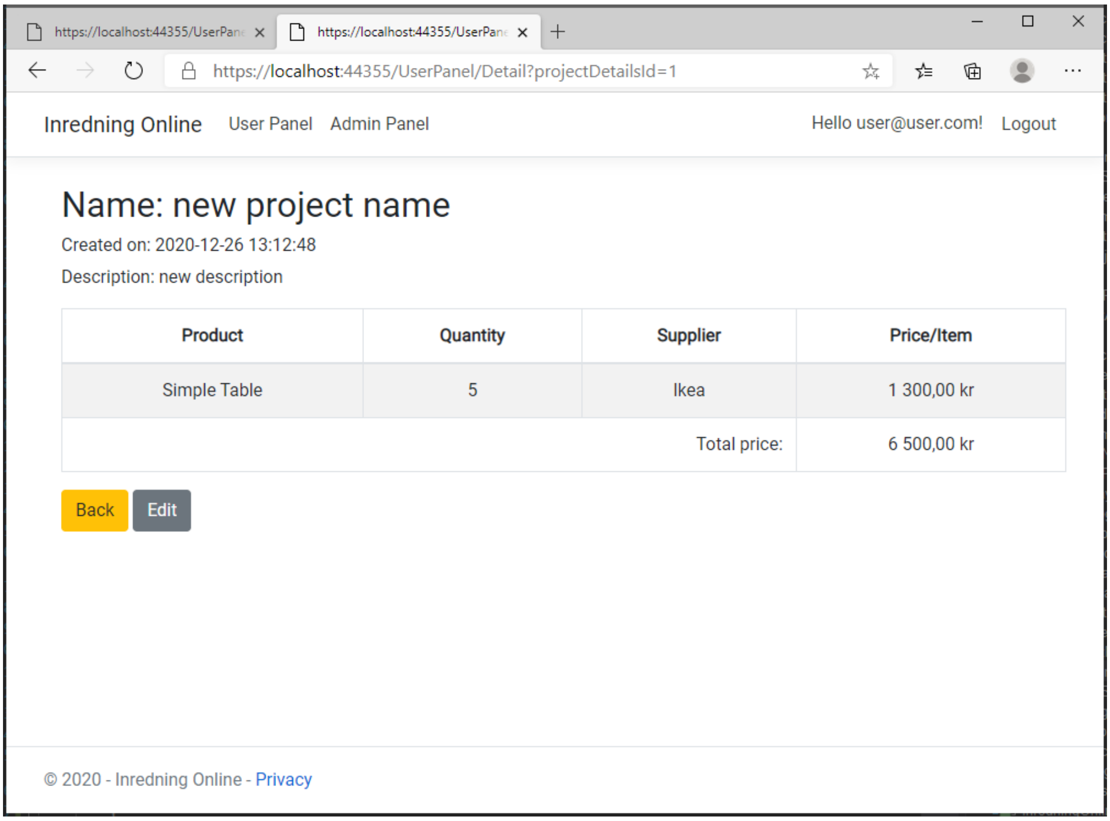
https://localhost:44355/UserPan
https://localhost:44355/UserPan
https://localhost:44355/UserPanel/Edit?projectDetailsId=1
Inredning Online
User Panel
Admin Panel
Hello user@user.com!
Logout
Update
© 2020 - Inredning Online - Privacy

Project Name: new project name

Description: new description

Quantity	Item	Supplier	Price/item
5	Simple Table	Ikea	1 300,00 kr

12. Changes after editing the project:



https://localhost:44355/UserPan
https://localhost:44355/UserPan
https://localhost:44355/UserPanel/Detail?projectDetailsId=1
Inredning Online
User Panel
Admin Panel
Hello user@user.com!
Logout
Back
Edit
© 2020 - Inredning Online - Privacy

Name: new project name

Created on: 2020-12-26 13:12:48

Description: new description

Product	Quantity	Supplier	Price/Item
Simple Table	5	Ikea	1 300,00 kr
		Total price:	6 500,00 kr