

# CodeIgniter MVC Activity: Product Catalog

## Objective

You will build a **Product Catalog web app** using CodeIgniter 4. The app should allow you to:

1. View all products
2. Add new products

Submit a working CodeIgniter Application:

- ProductModel.php
- Products.php controller
- index.php and create.php views

Also include screenshots showing the product list and adding a new product.

This activity will help you understand the MVC (Model-View-Controller) structure in CodeIgniter.

## Step 1 — Database Setup

1. Create a database (example: ci\_products).
2. Inside that database, run this SQL:

```
CREATE TABLE products (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  price DECIMAL(10,2) NOT NULL,  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

## Step 2 — Define Routes

Open the file: app/Config/Routes.php and add these lines:

```
$routes->get('products', 'Products::index');  
$routes->get('products/create', 'Products::create');  
$routes->post('products/store', 'Products::store');
```

### Step 3 — Create the Model

File: app/Models/ProductModel.php

```
<?php
namespace App\Models;
use CodeIgniter\Model;

class ProductModel extends Model
{
    protected $table = 'products';
    protected $primaryKey = 'id';
    protected $allowedFields = ['name', 'price'];
    public $timestamps = false;
}
```

### Step 4 — Create the Controller

File: app/Controllers/Products.php

```
<?php
namespace App\Controllers;
use App\Models\ProductModel;
class Products extends BaseController
{
    protected $productModel;
    public function __construct()
    {
        $this->productModel = new ProductModel();
    }
    public function index()
    {
        $data['products'] = $this->productModel->findAll();
        return view('products/index', $data);
    }
    public function create()
    {
        return view('products/create');
    }
    public function store()
    {
        $this->productModel->save([
            'name' => $this->request->getPost('name'),
            'price' => $this->request->getPost('price')
        ]);
        return redirect()->to('/products')->with('message', 'Product added successfully!');
    }
}
```

## Step 5 — Create the Views

Create folder: app/Views/products/

### One File for: index.php

```
<h1>Product Catalog</h1>

<?php if (session()->getFlashdata('message')): ?>
    <p style="color:green"><?= session()->getFlashdata('message') ?></p>
<?php endif; ?>

<a href="/products/create">+ Add Product</a>
<ul>
<?php foreach ($products as $p): ?>
    <li><?= esc($p['name']) ?> - ₪<?= esc($p['price']) ?></li>
<?php endforeach; ?>
</ul>
```

### Another File for: create.php

```
<h1>Add New Product</h1>
<form action="/products/store" method="post">
    <label>Product Name:</label><br>
    <input type="text" name="name" required><br><br>

    <label>Price:</label><br>
    <input type="number" step="0.01" name="price" required><br><br>

    <button type="submit">Save</button>
</form>
```

## Test Your App

1. Open <http://localhost:8080/products> → You should see a list of products.
2. Click Add Product → Fill form → Submit.
3. The new product should appear in the list.

## Additional Challenge (20 Points)

- Add form validation (e.g., price must be greater than 0).
- Add a delete button to remove products.
- Style your pages with Bootstrap.