

Practical Nest.js Book Images

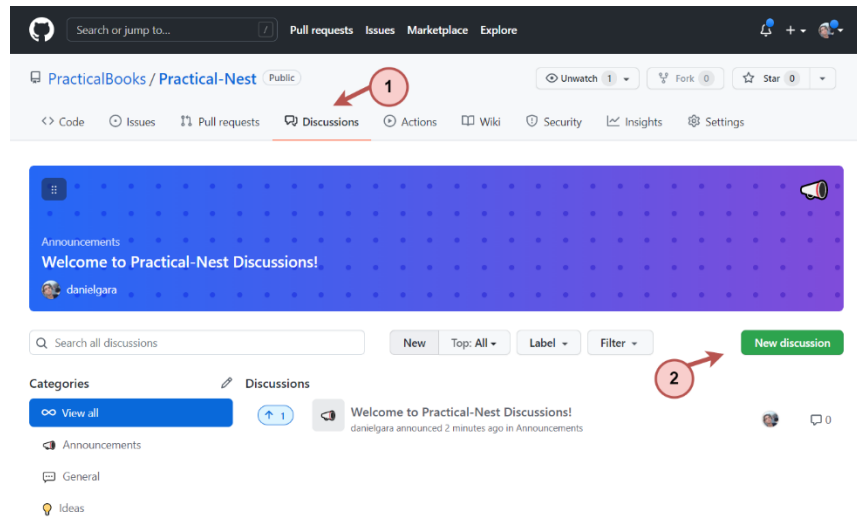


Figure P-1. Discussion zone of the GitHub repository.

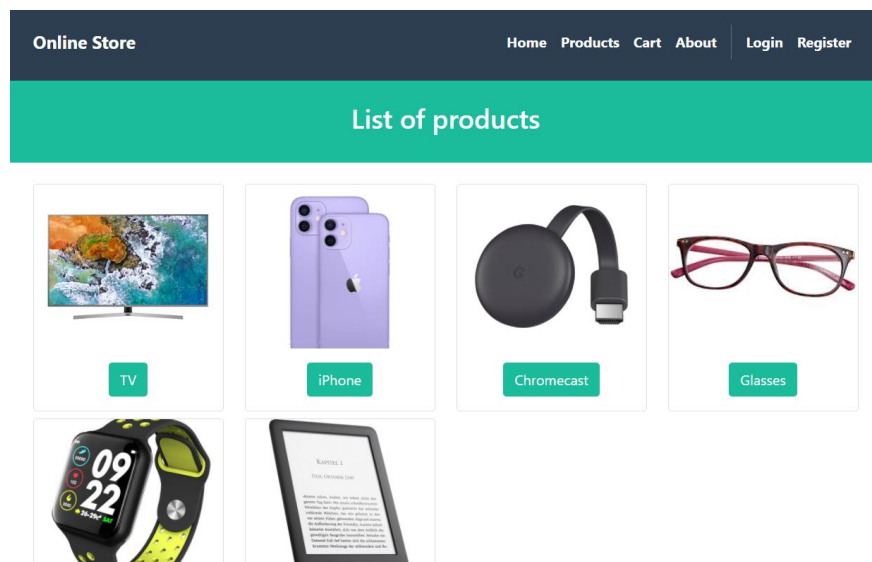


Figure 1-1. List of products page.

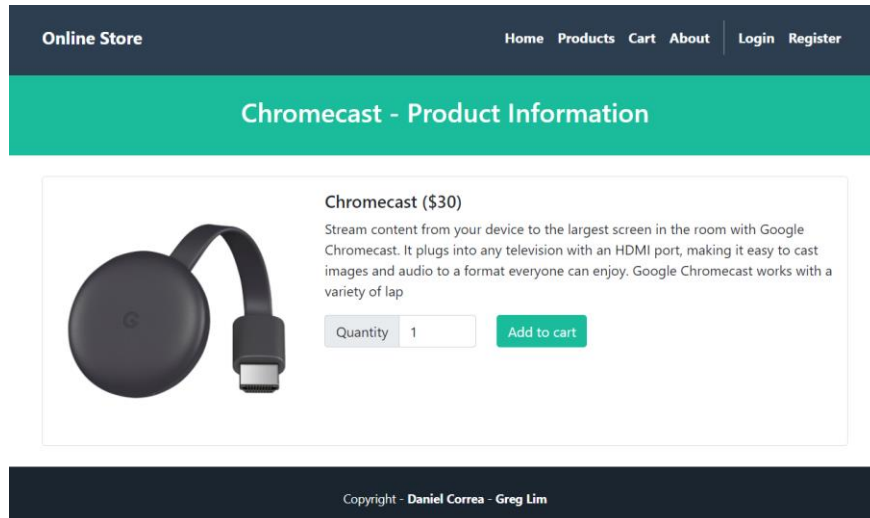


Figure 1-2. Product page.

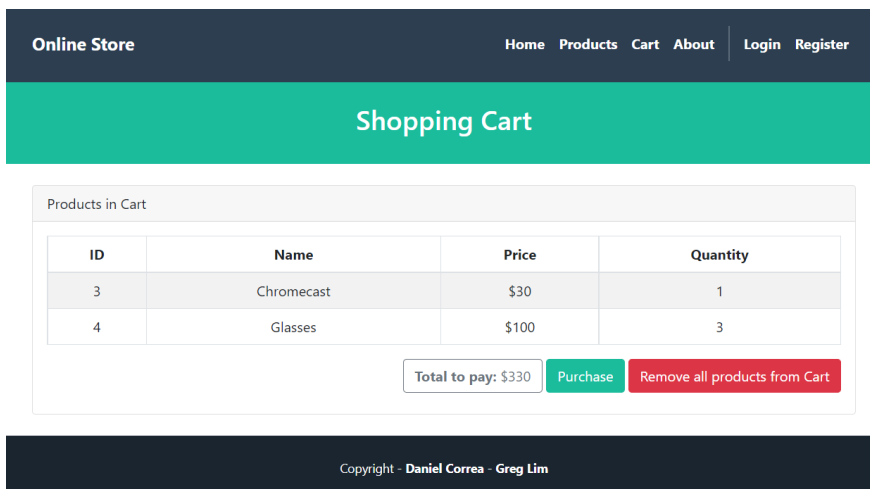


Figure 1-3. Shopping cart page.

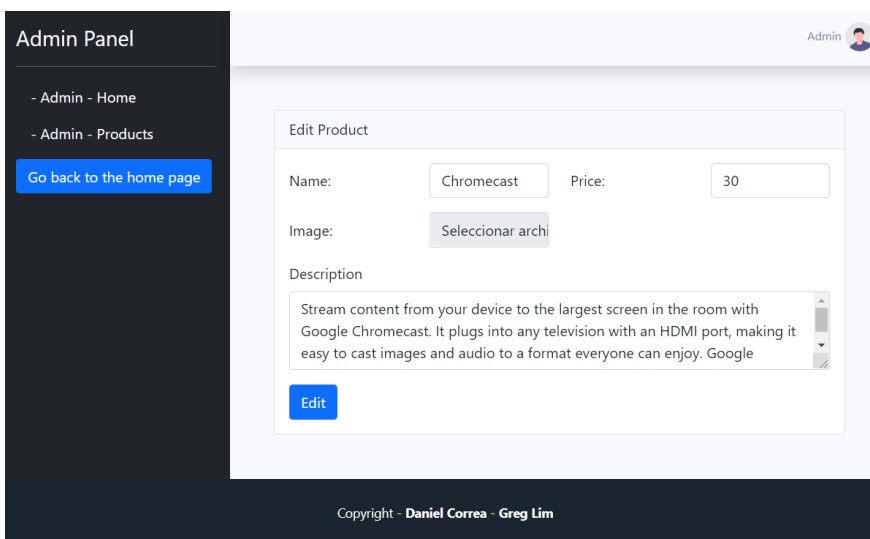


Figure 1-4. Admin panel page.

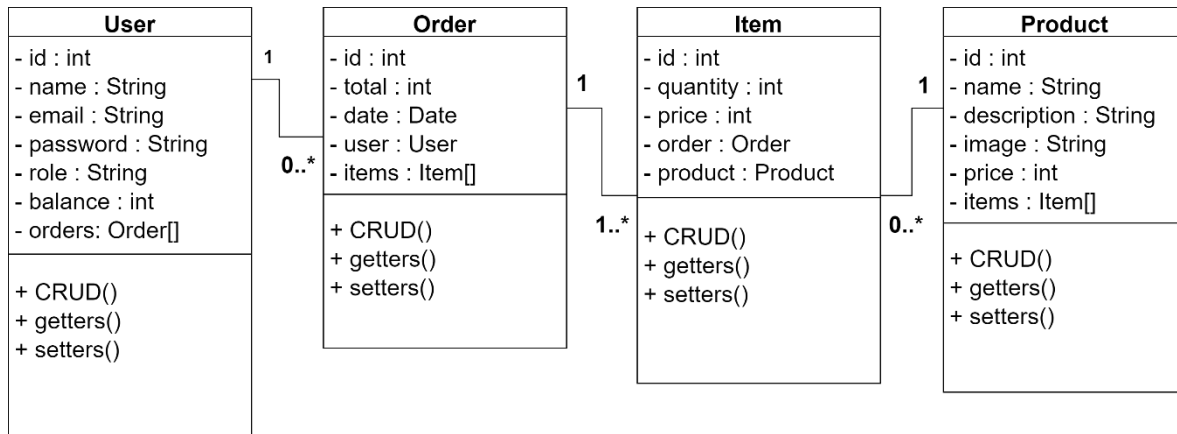


Figure 2-1. Online Store class diagram.

```
PS C:\Users\yo> node -v
v16.13.1
```

Figure 3-1. Checking Node.js version.

```
PS C:\Users\yo> npm -v
8.3.0
```

Figure 3-2. Checking composer version.

```
PS C:\xampp\htdocs\online-store> npm run start

> online-store@0.0.1 start
> nest start

[Nest] 21428 - 12/12/2021, 5:45:49 p. m. LOG [NestFactory] Starting Nest application...
[Nest] 21428 - 12/12/2021, 5:45:49 p. m. LOG [InstanceLoader] AppModule dependencies initialized +29ms
[Nest] 21428 - 12/12/2021, 5:45:49 p. m. LOG [RoutesResolver] AppController {/}: +6ms
[Nest] 21428 - 12/12/2021, 5:45:49 p. m. LOG [RouterExplorer] Mapped {/, GET} route +3ms
[Nest] 21428 - 12/12/2021, 5:45:49 p. m. LOG [NestApplication] Nest application successfully started +2ms
```

Figure 3-3. Running Nest project.

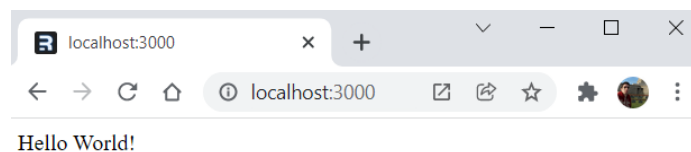


Figure 3-4. Nest default main page.

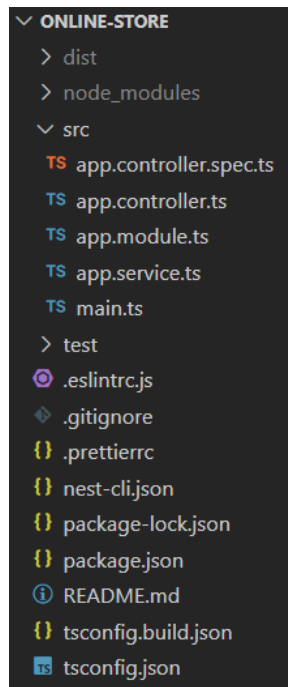


Figure 3-5. Nest project structure.

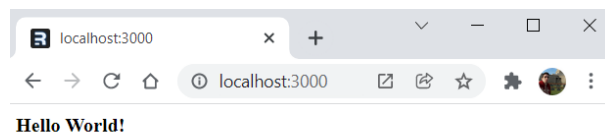


Figure 3-6. Nest main page.

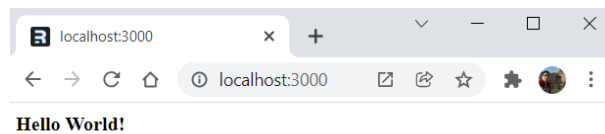


Figure 4-1. Online Store index page with Handlebars.

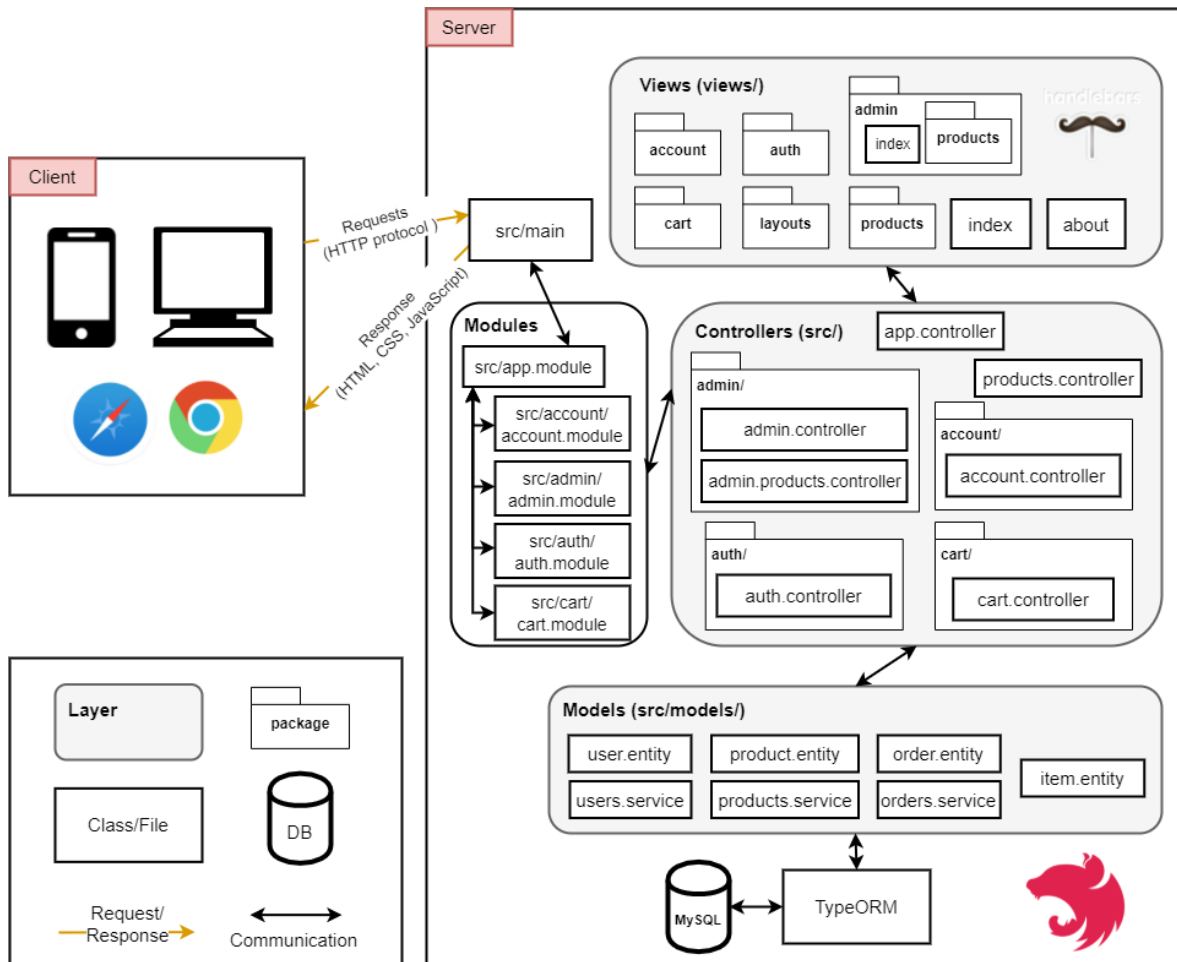


Figure 5-1. Online Store software architecture.

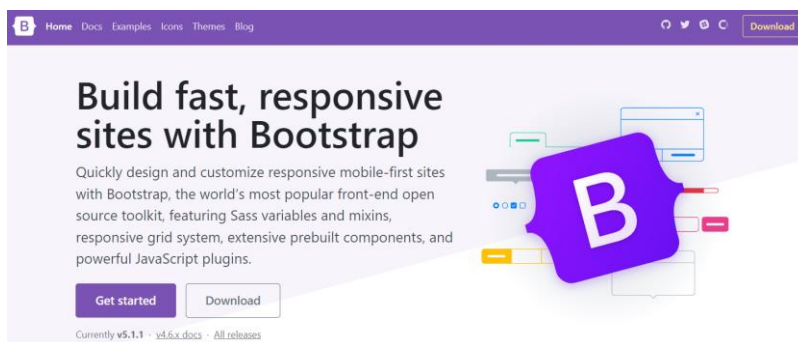


Figure 6-1. Bootstrap website.

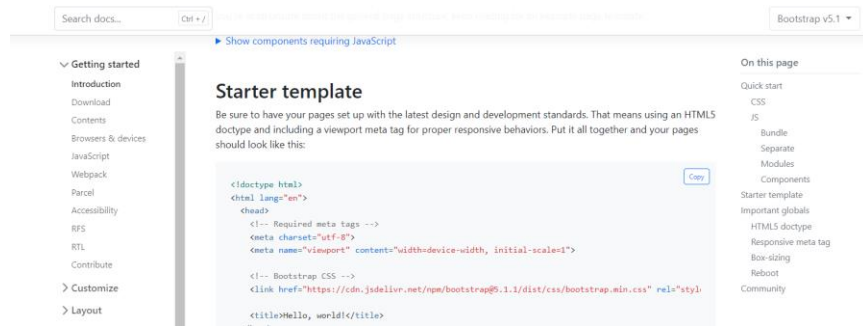


Figure 6-2. Bootstrap starter template.

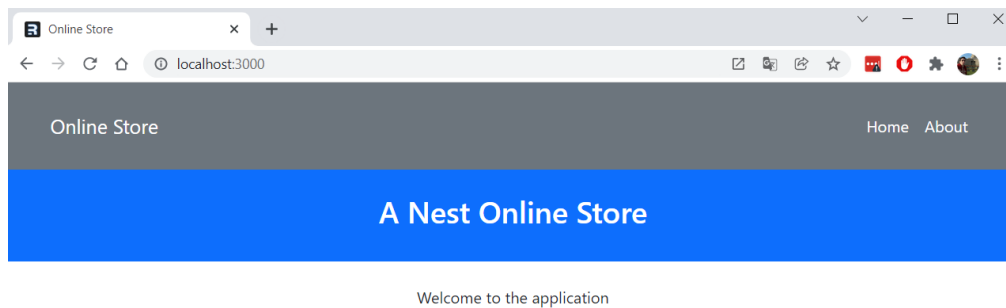


Figure 6-3. Application home page with layout.

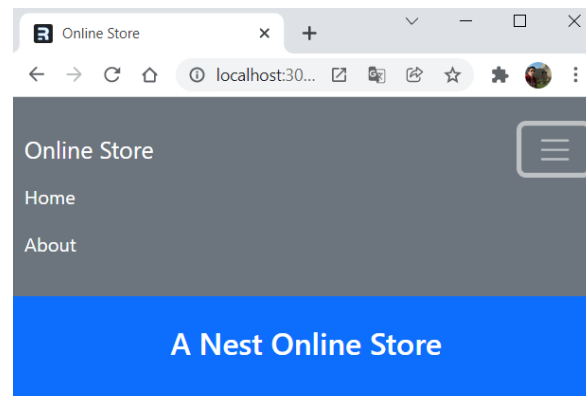


Figure 6-4. Application home page with reduced browser window width.

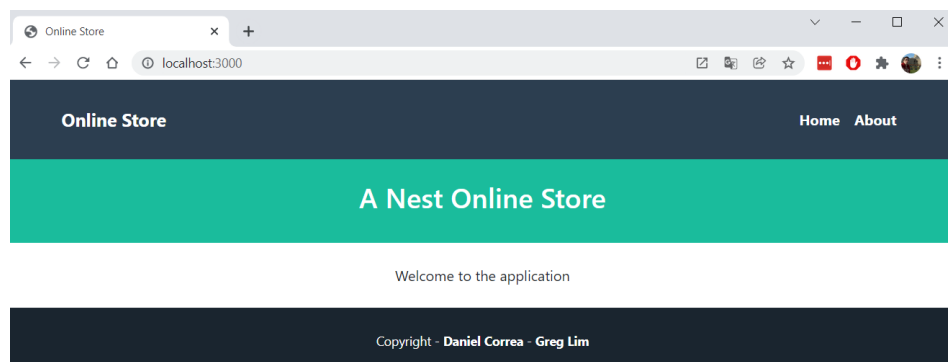


Figure 6-5. Application home page with refined layout.

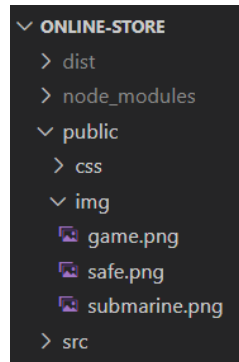


Figure 7-1. Project structure after storing the images.

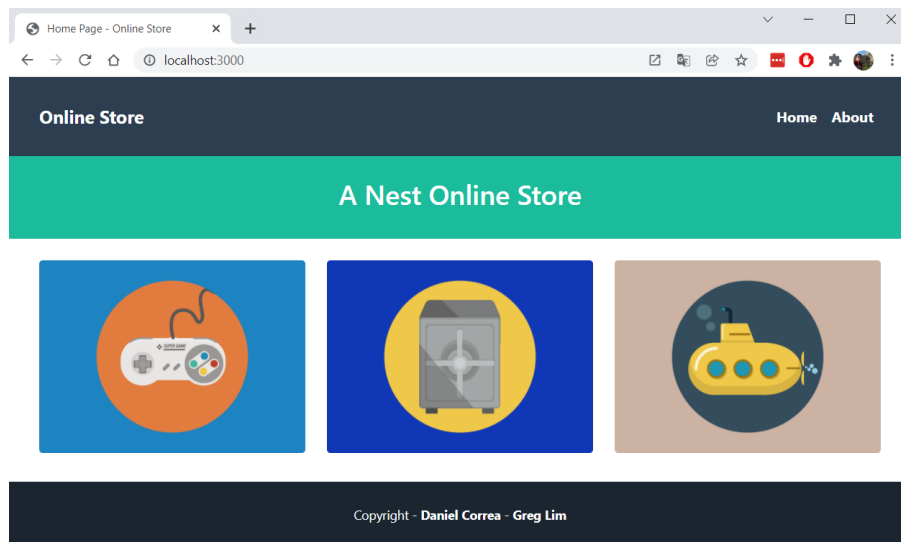


Figure 7-2. Online Store – Home page.

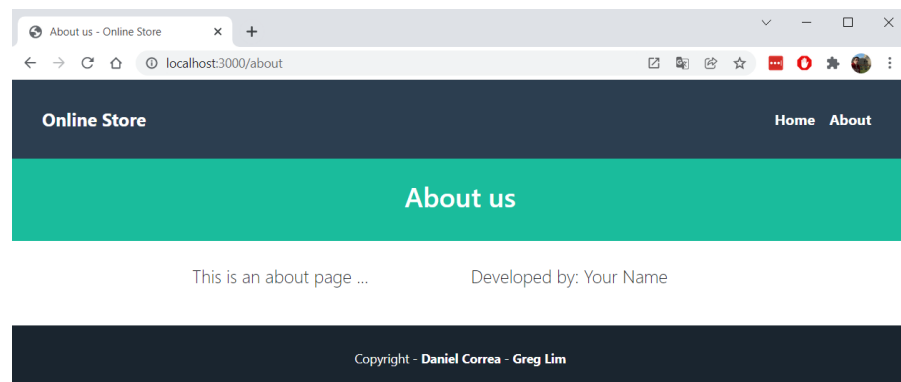


Figure 7-3. Online Store – About page.

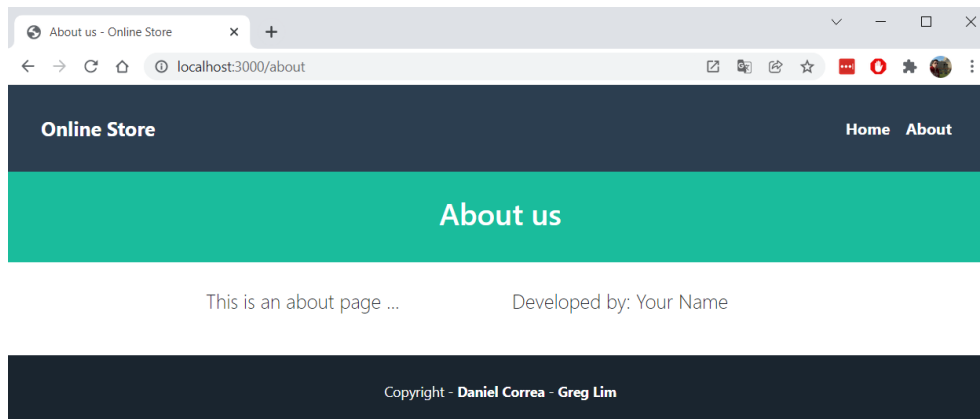


Figure 8-1. Online Store – About page.

```
PS C:\xampp\htdocs\online-store> npm run lint
> online-store@0.0.1 lint
> eslint "{src,apps,libs,test}/**/*.ts" --fix
```

Figure 9-1. Running ESLint.

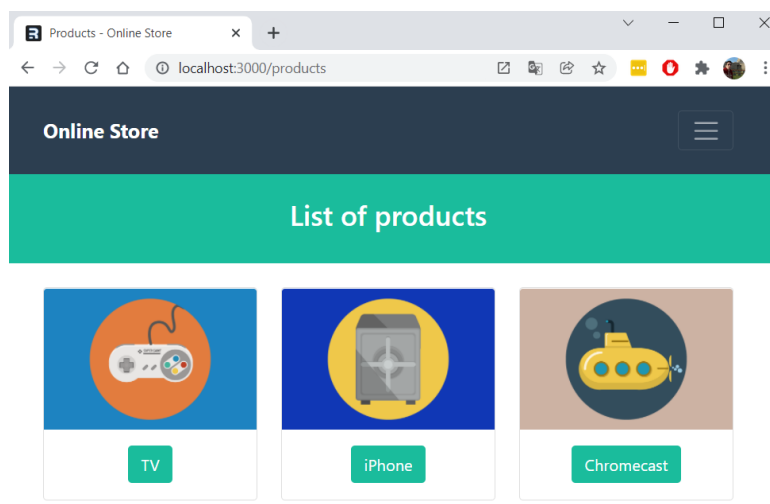


Figure 10-1. Online Store – Products page.

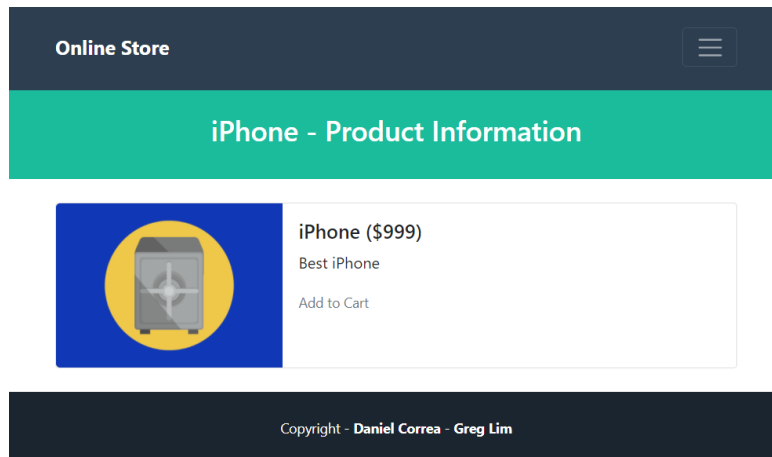


Figure 10-2. Online Store – iPhone product page.

id	name	description	image	price
1	TV	Best tv	game.png	1000
2	iPhone	Best iPhone	safe.png	999
3	Chromecast	Best Chromecast	submarine.png	30
4	Glasses	Best Glasses	game.png	100

Figure 11-1. Product table.

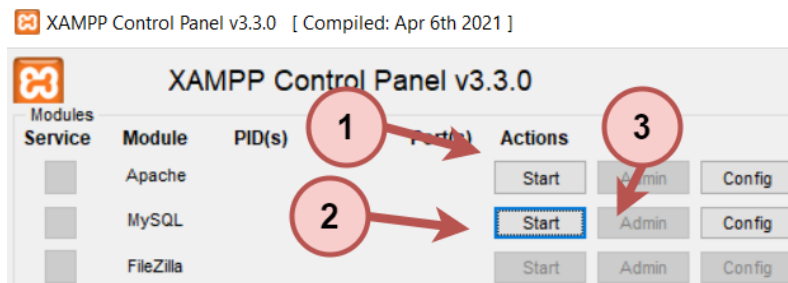


Figure 11-2. Starting MySQL module in XAMPP.

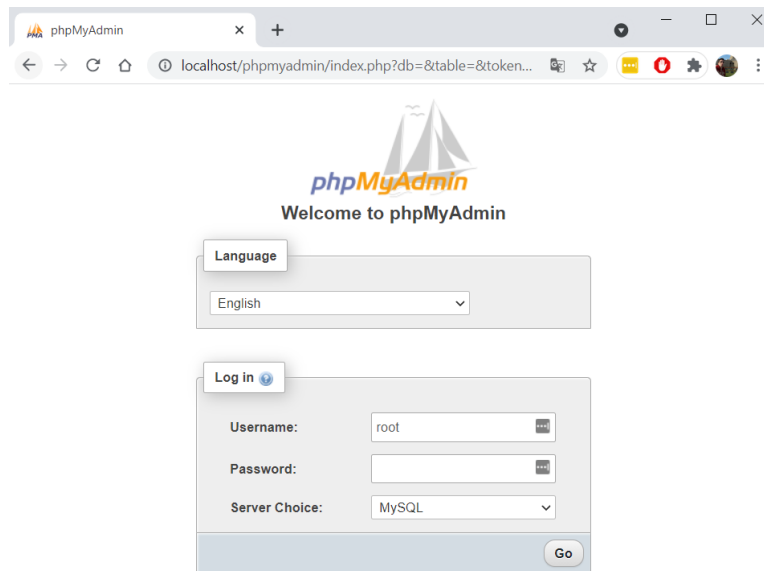


Figure 11-3. XAMPP phpMyAdmin application.

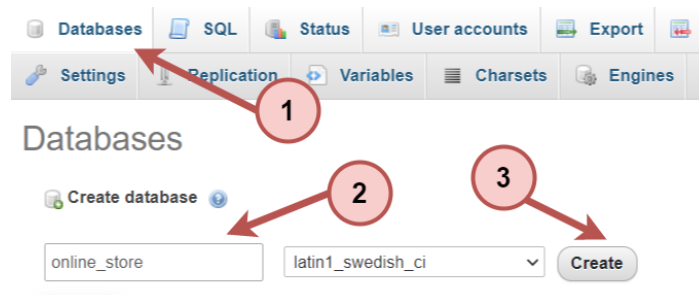


Figure 11-4. Database creation.

```
[7:02:22 p. m.] Starting compilation in watch mode...
[7:02:24 p. m.] Found 0 errors. Watching for file changes.

[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [NestFactory] Starting Nest application...
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [InstanceLoader] TypeOrmModule dependencies initialized +139ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [InstanceLoader] AppModule dependencies initialized +1ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [InstanceLoader] TypeOrmCoreModule dependencies initialized +34ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RoutesResolver] AppController {/}: +6ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RouterExplorer] Mapped {/, GET} route +3ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RouterExplorer] Mapped {/about, GET} route +1ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RoutesResolver] ProductsController {/products}: +0ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RouterExplorer] Mapped {/products, GET} route +1ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [RouterExplorer] Mapped {/products/:id, GET} route +1ms
[Nest] 18856 - 06/01/2022, 7:02:25 p. m. LOG [NestApplication] Nest application successfully started +2ms
```

Figure 12-1. TypeORM proper configured.

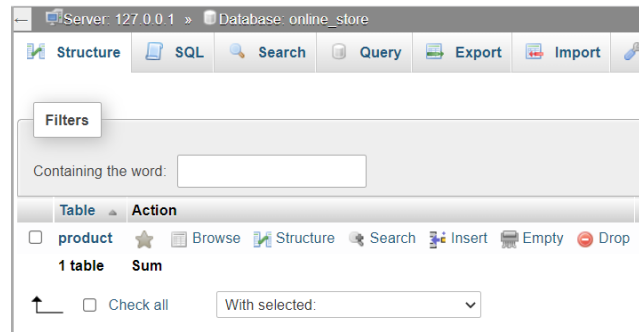


Figure 13-1. New product table – phpMyAdmin.

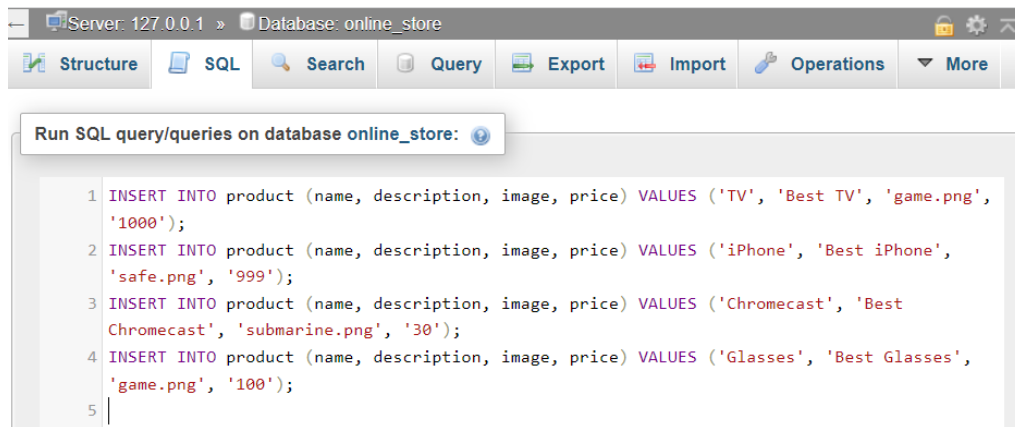


Figure 13-2. Inserting products through phpMyAdmin.

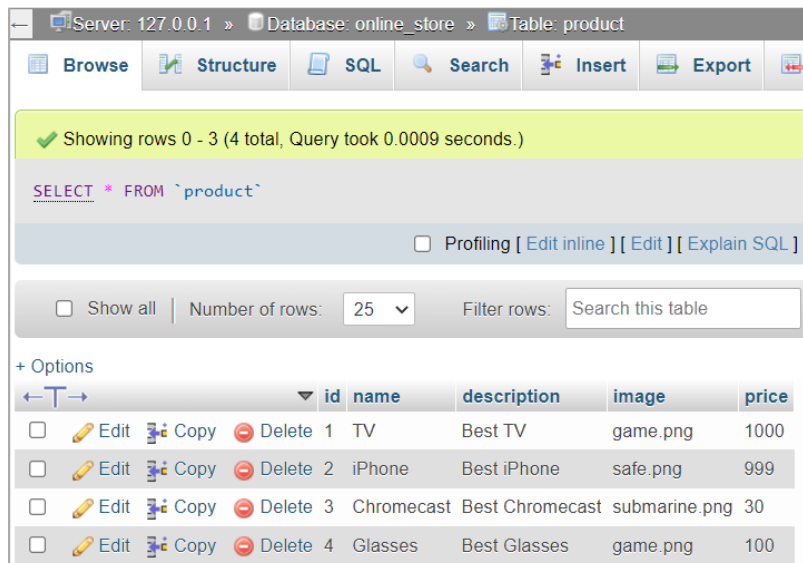


Figure 13-3. Products displayed in phpMyAdmin.

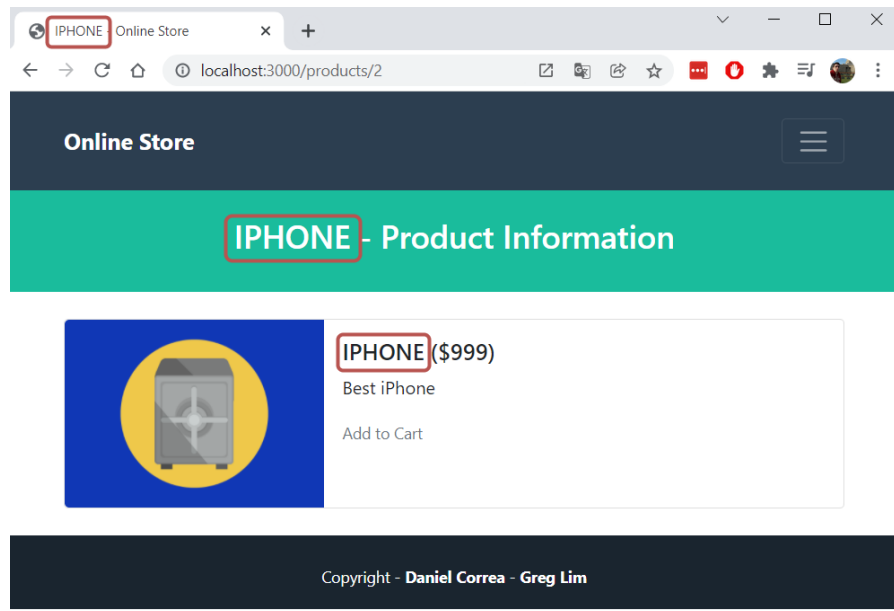


Figure 15-1. Accessing a product with the modified *getName* method.

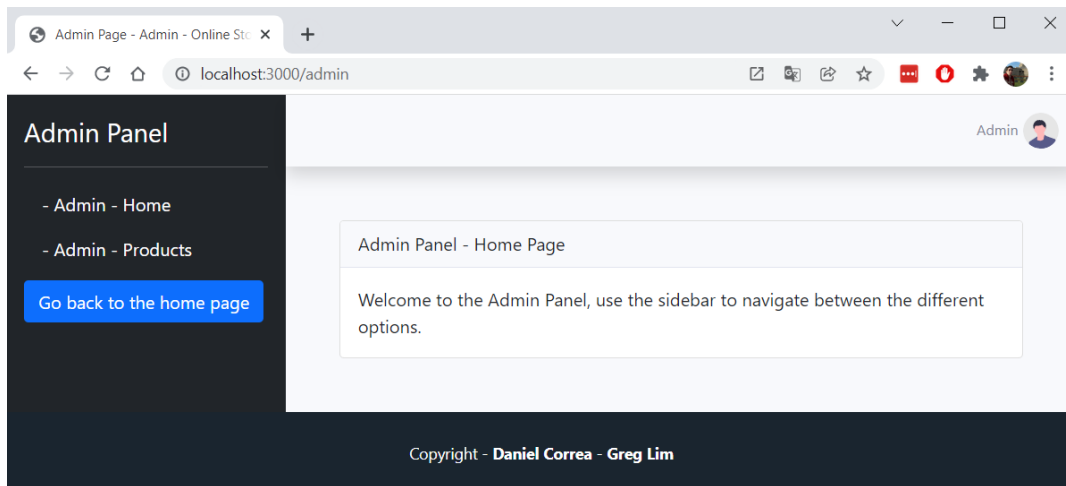


Figure 16-1. Online Store – Admin Panel.

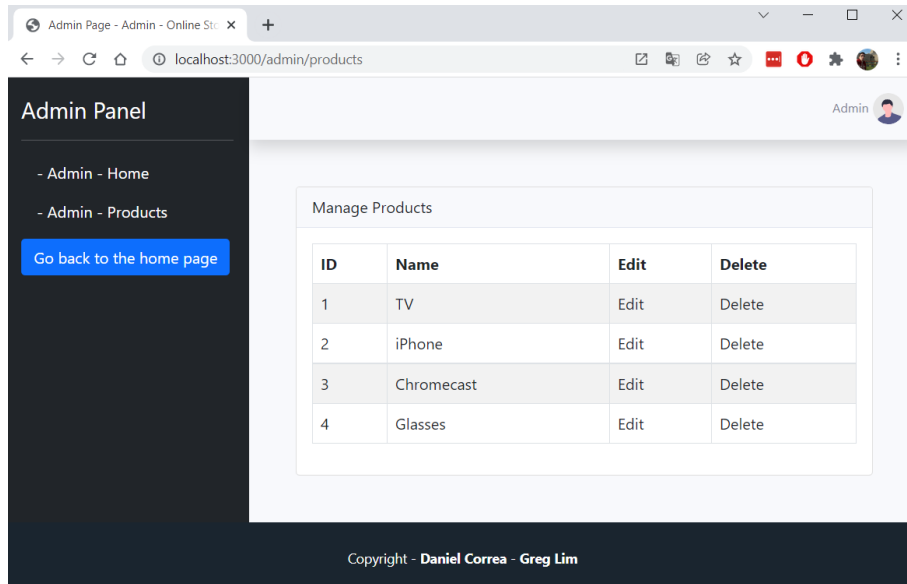


Figure 17-1. Online Store – Admin Panel – Products.

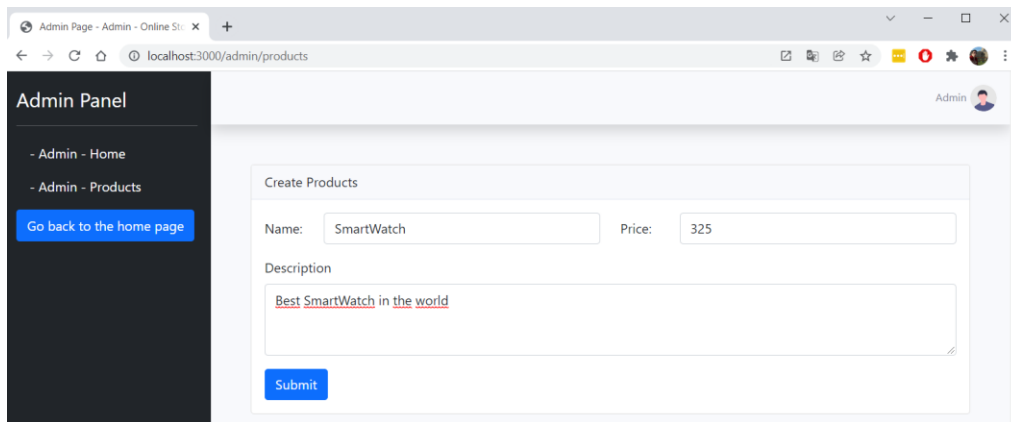


Figure 18-1. Online Store – Refined Admin Panel – Products.

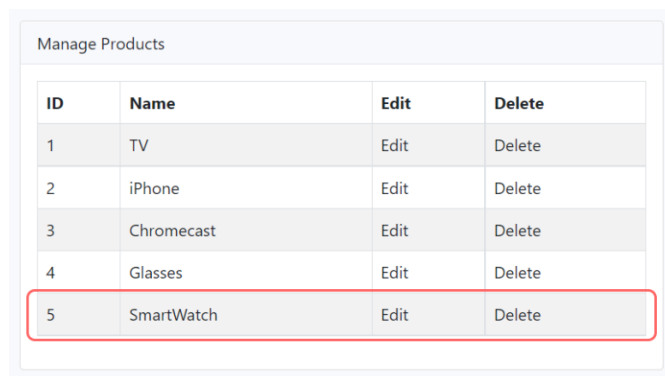



Figure 18-2. New product listed.

Admin Panel

- Admin - Home
- Admin - Products

[Go back to the home page](#)

Admin 

Create Products

Name: Price:

Image:

Description

Figure 19-1. Online Store – Create products form with image selection.

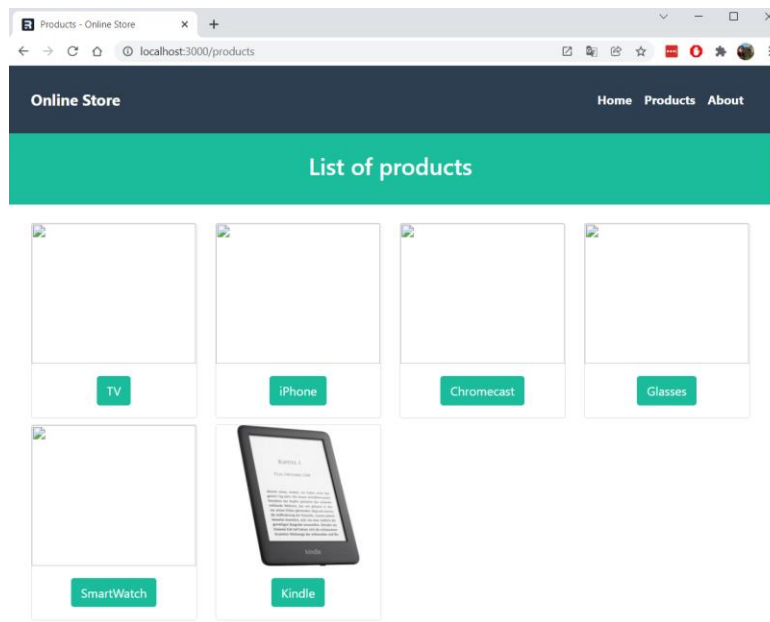


Figure 19-2. Online Store – List products with their uploaded images.













ID	Name	Edit	Delete
1	TV		
2	iPhone		
3	Chromecast		
4	Glasses		
5	SmartWatch		
6	Kindle		

Figure 20-1. Online Store – Refined Admin Panel – Products.

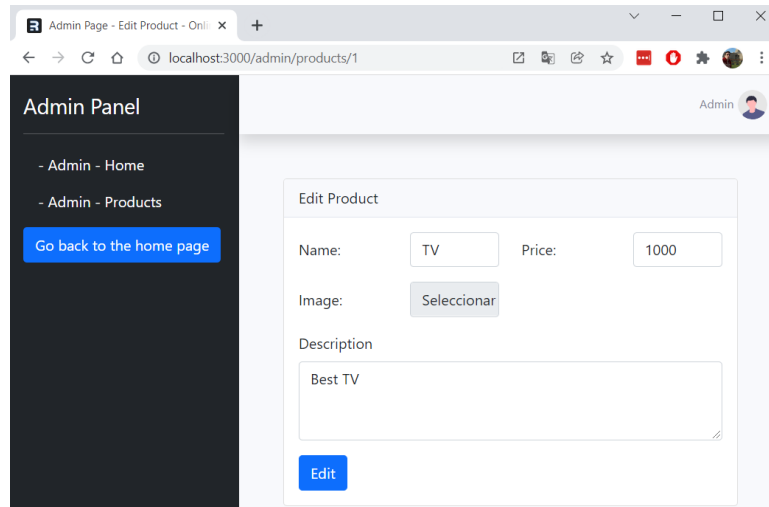


Figure 20-2. Editing a product.

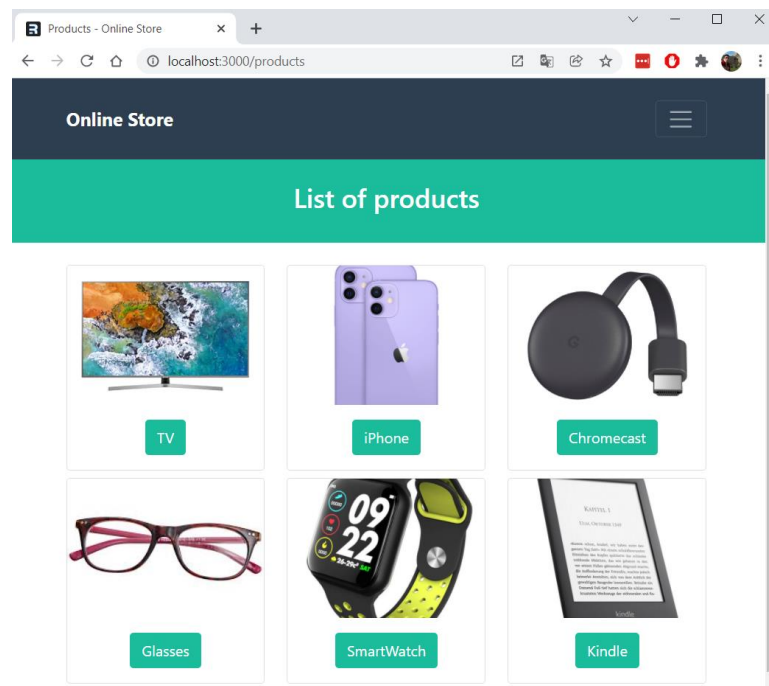


Figure 20-3. List products with proper images.

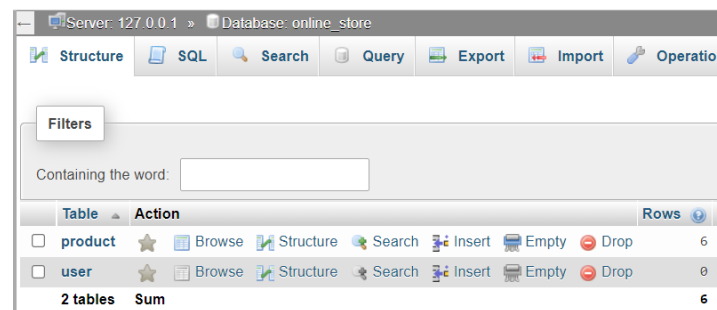


Figure 21-1. New user table – phpMyAdmin.

Online Store

HomeProductsAboutRegister

User Register

Register

Name

Email

Password

Register

Copyright - Daniel Correa - Greg Lim

Figure 21-2. Online Store – Registration form.

id	name	email	password	role	balance
1	Daniel	books@danielgara.com	\$2b\$10\$voCE1cYClzpLIMFHB.FckOZZ5oyv0U5FUIMd4am2Ges...	client	1000

Figure 21-3. New user row – phpMyAdmin.

Online Store

HomeProductsAboutLoginRegister

User Login

Login

Email

Password

Login

Copyright - Daniel Correa - Greg Lim

Figure 22-1. Online Store – Login form.

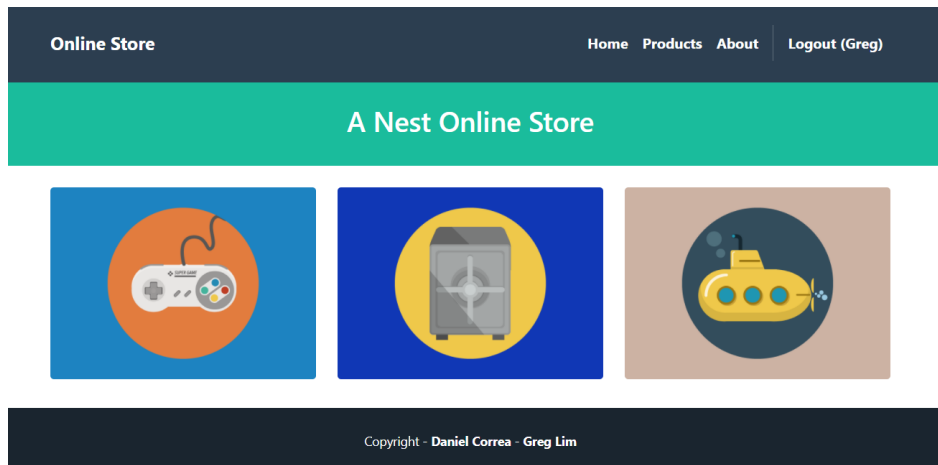


Figure 22-2. Online Store – Home page for a logged user.

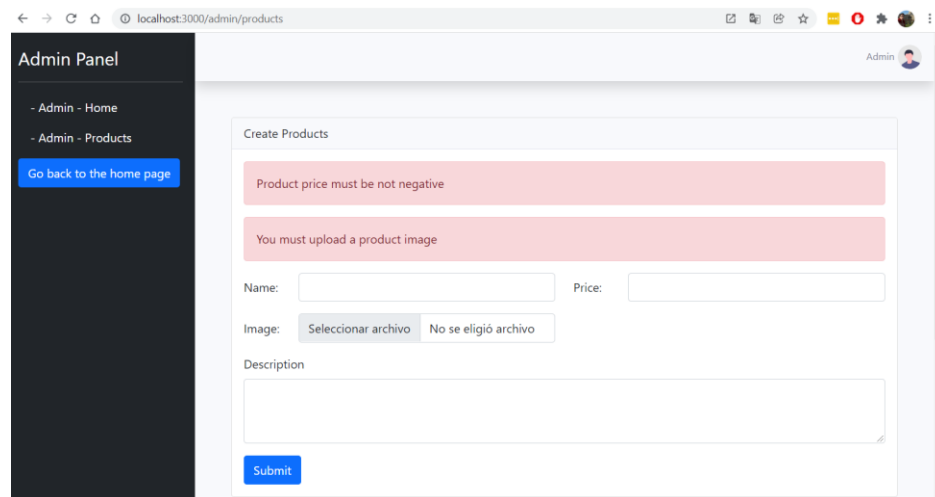


Figure 23-1. Online Store – Creating a product with invalid data.

id	name	email	password	role	balance
1	Daniel	books@danielgara.com	\$2b\$10\$bWnAu8QiQLUGSkP0KEjl/eg.qesLvTeX6w2qEF6JtVJ...	admin	1000
2	Greg	support@i-ducate.com	\$2b\$10\$TCA47PMCIXECm.HOCrSbF.E..jpXzWVAGeBY1Bko1f7...	client	1000

Figure 24-1. Editing a user's role.

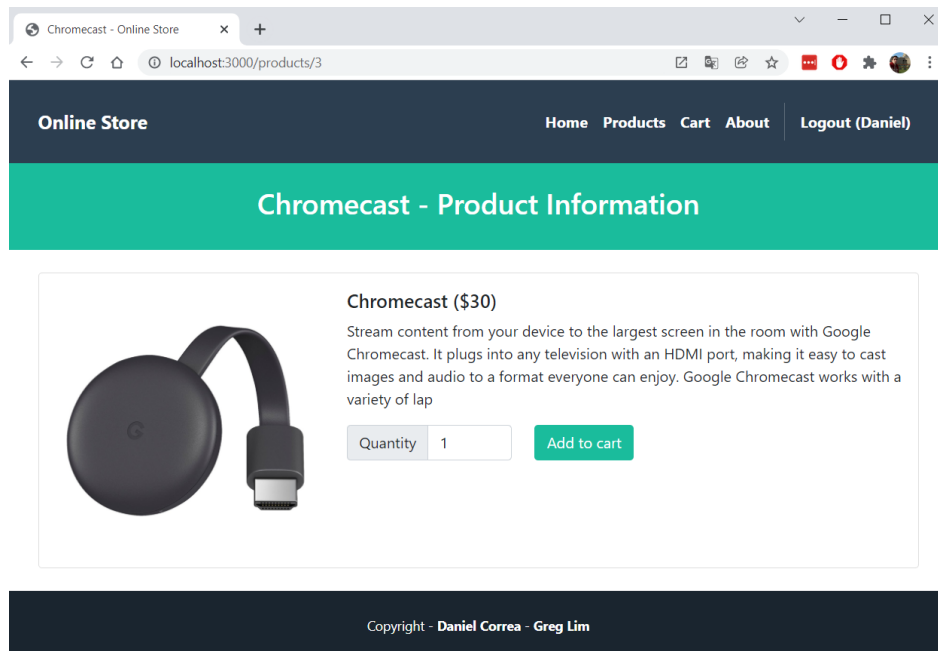


Figure 25-1. Product show view with add to cart button.

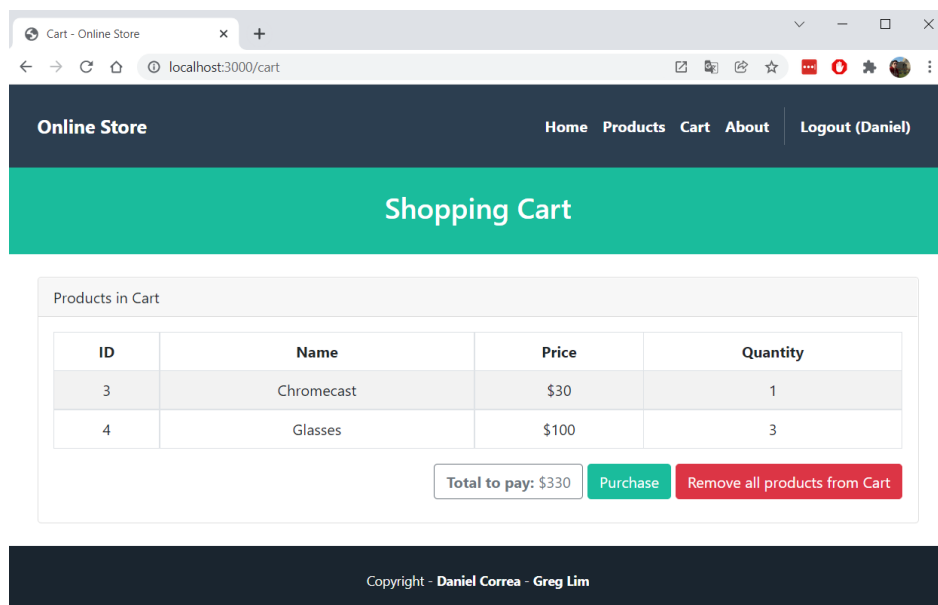


Figure 25-2. Online Store – Shopping cart page.

Order #1

Date: 2021-10-14

Total: \$1060

User id - name: 1 - Daniel

Qty	Product id - name	Price
1	1 - TV	1000
2	3 - Chromecast	30

Figure 26-1. Example of an invoice.

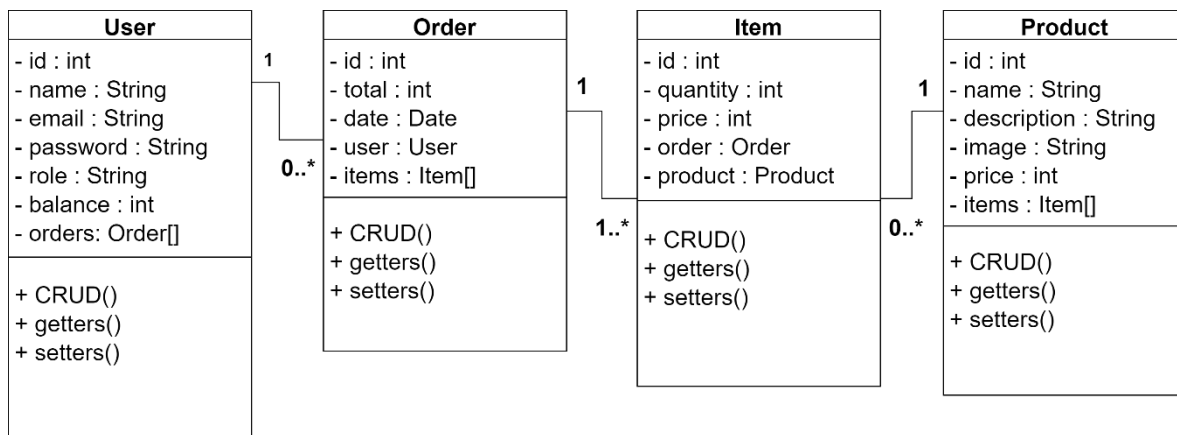


Figure 26-2. Original Online Store class diagram.

Server: 127.0.0.1 » Database: online_store

Structure SQL Search Query Export Import Operations

Filters

Containing the word:

Table	Action	Rows
<input type="checkbox"/> item	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> order	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> product	★ Browse Structure Search Insert Empty Drop	6
<input type="checkbox"/> user	★ Browse Structure Search Insert Empty Drop	2
4 tables	Sum	8

Figure 26-3. New order and item tables – phpMyAdmin.

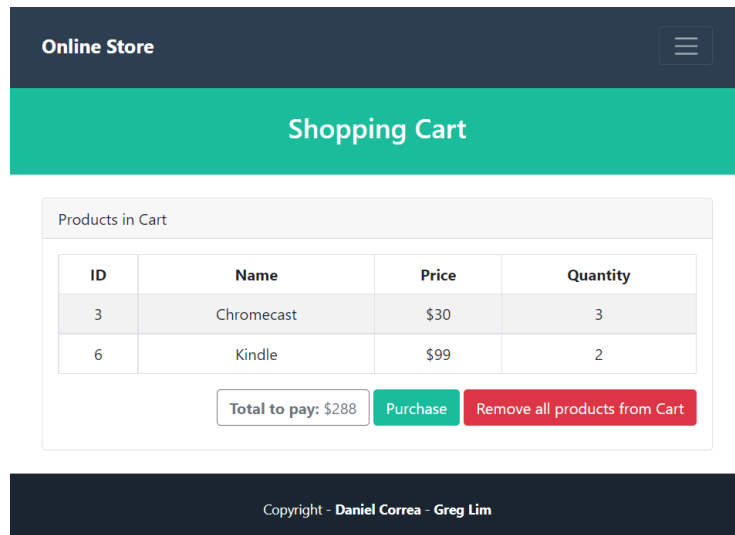


Figure 27-1. Shopping cart with some products.

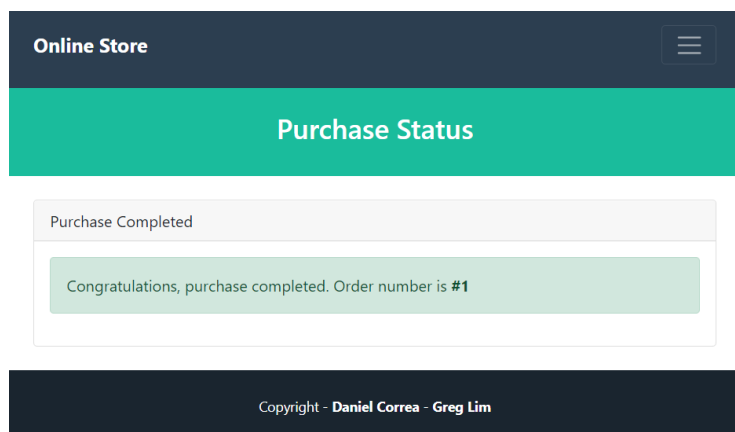


Figure 27-2. Purchase completed message.

id	total	date	userId
1	288	2022-01-13 14:51:41.423611	1

Figure 27-3. New order row included in the database.

id	quantity	price	orderId	productId
1	3	30	1	3
2	2	99	1	6

Figure 27-4. New item rows included in the database.

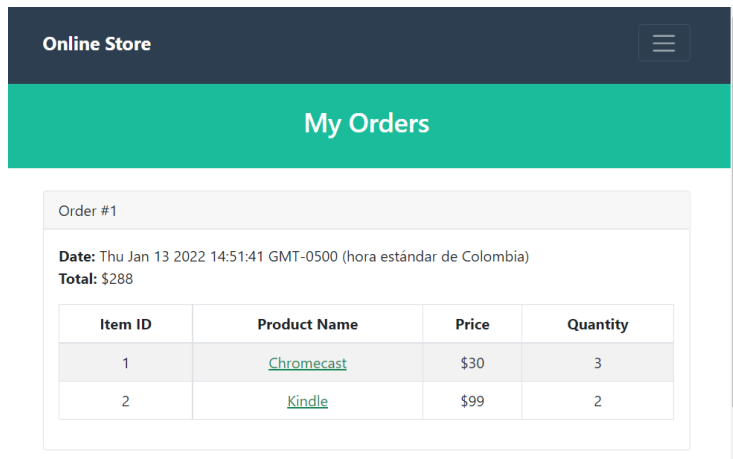


Figure 28-1. Online Store – My Orders page.

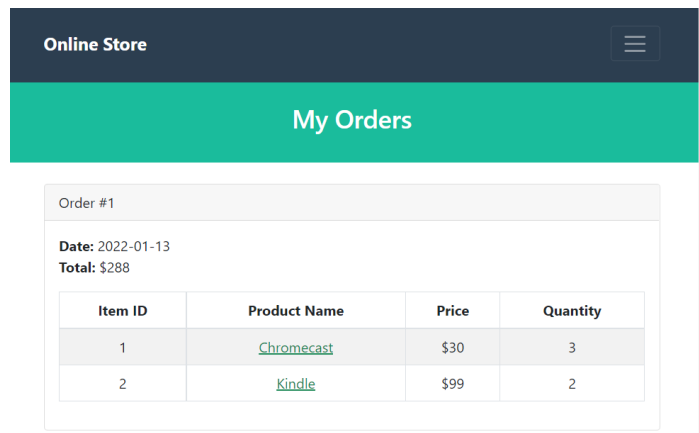


Figure 28-2. Online Store – Orders page with new date format.

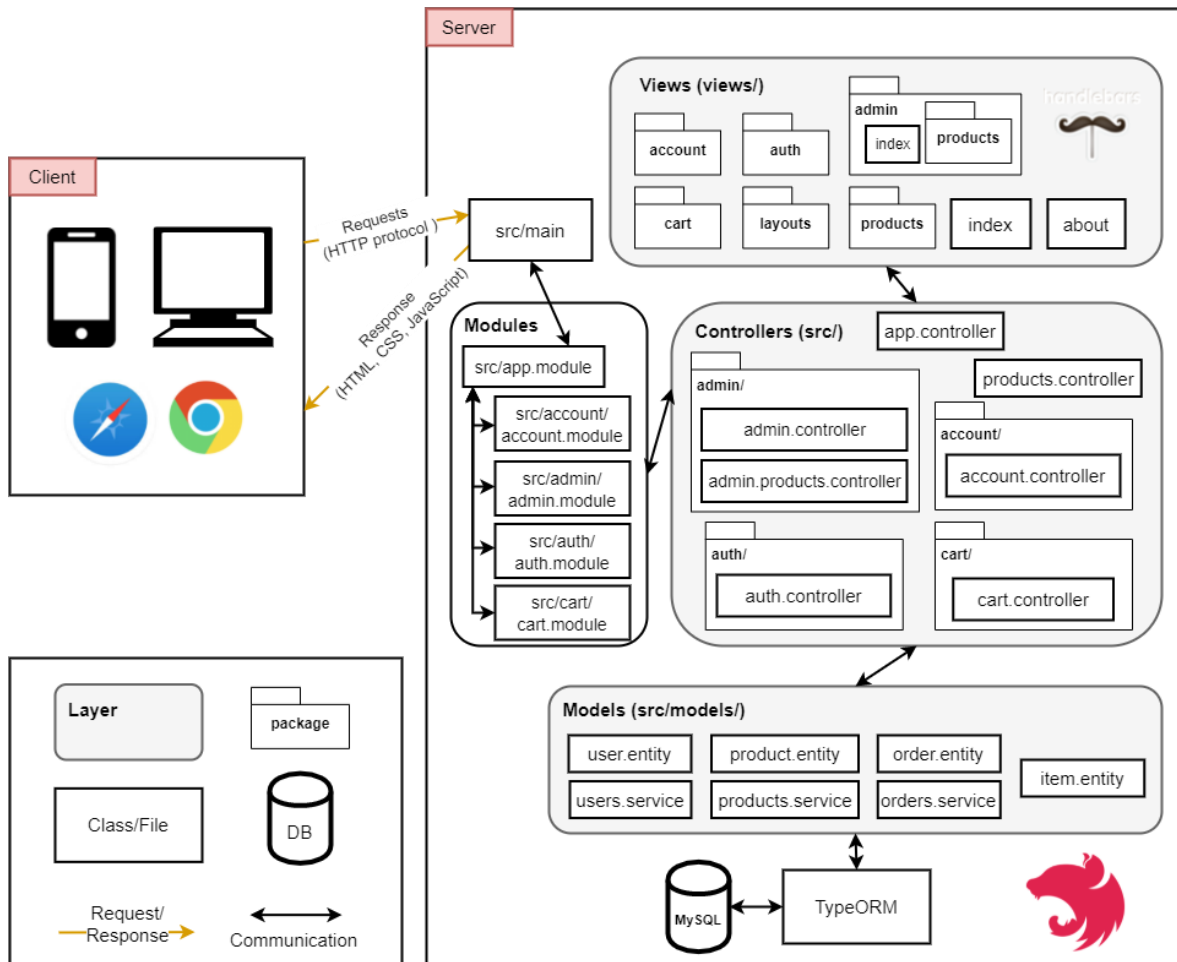


Figure 28-3. Online Store software architecture.

The image shows a "Sign up with Email" form on a light gray background. The form includes the following elements:

- Title:** "Sign up with Email"
- EMAIL ADDRESS:** A text input field containing "books@danielgara.com".
- PASSWORD:** A text input field with masked characters "*****".
- Terms & Conditions:** A checkbox that is checked, followed by the text "By clicking this I have read and agree to the [Terms & Conditions](#)".
- Submit Button:** A blue button labeled "SIGN UP WITH EMAIL".

Figure 29-1. Clever Cloud Sign up page.

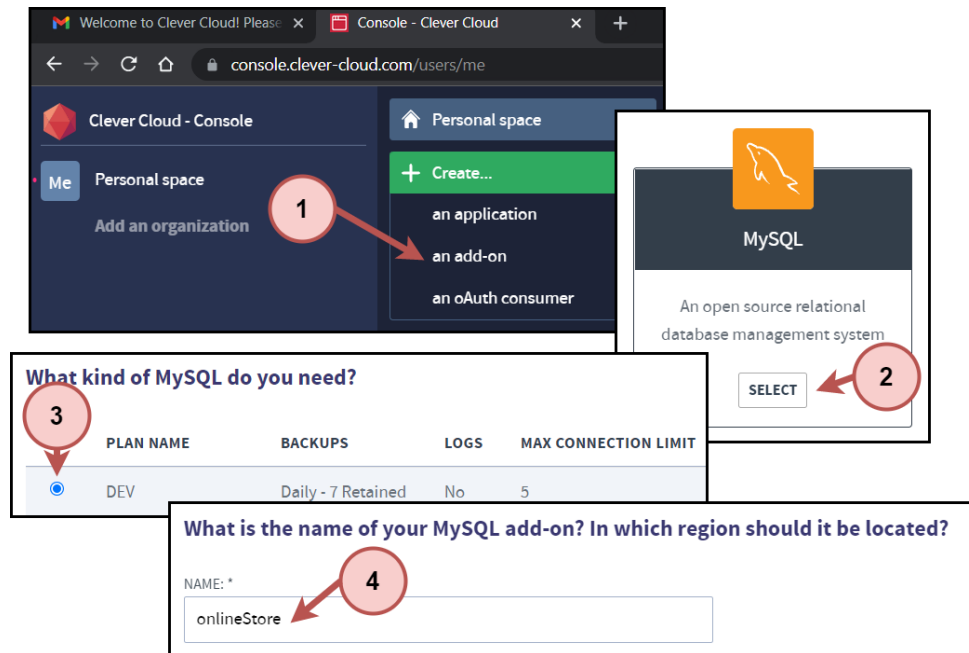


Figure 29-2. Clever Cloud MySQL add-on creation.

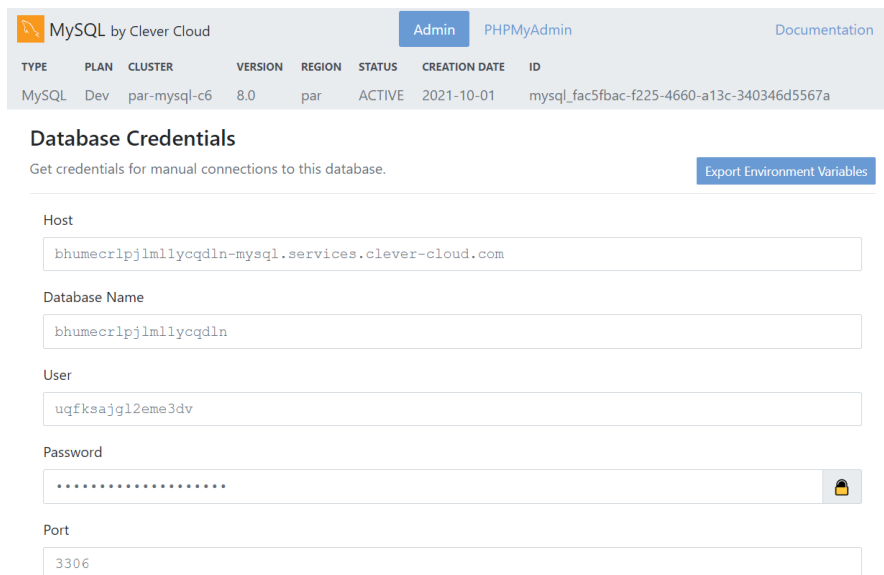


Figure 29-3. Clever Cloud MySQL add-on creation.

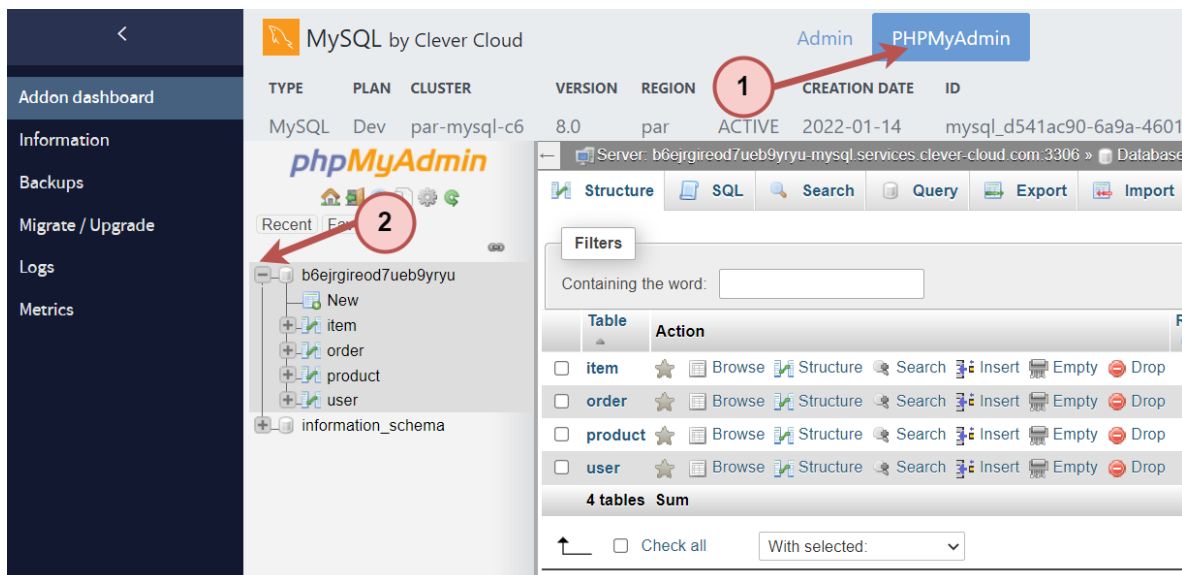


Figure 29-4. Accessing phpMyAdmin over the Clever Cloud database.

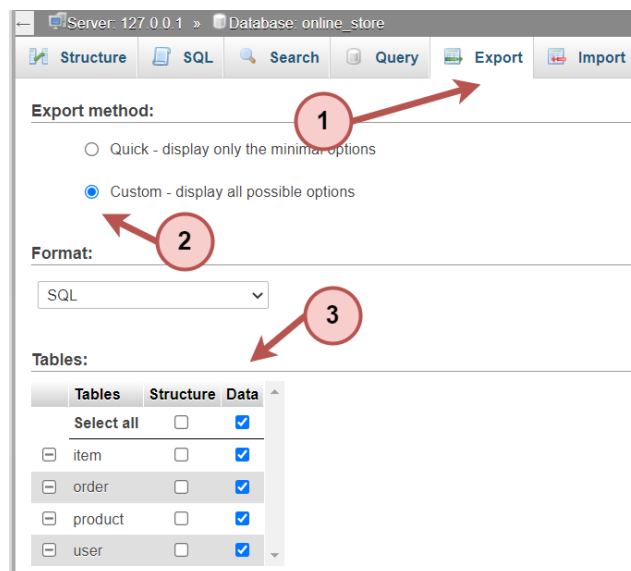


Figure 29-5. Exporting the data from the local database.

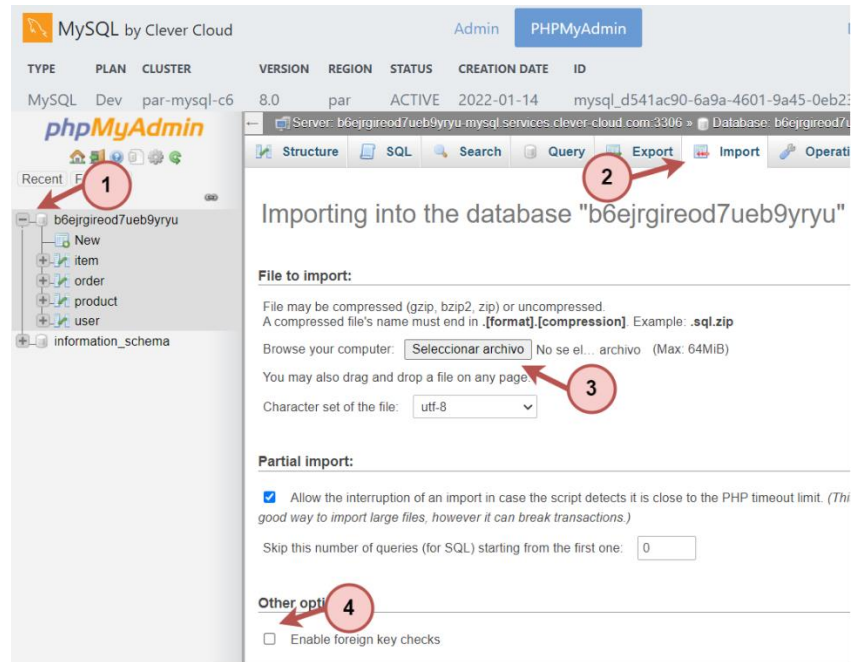


Figure 29-6. Importing the data to the cloud database.

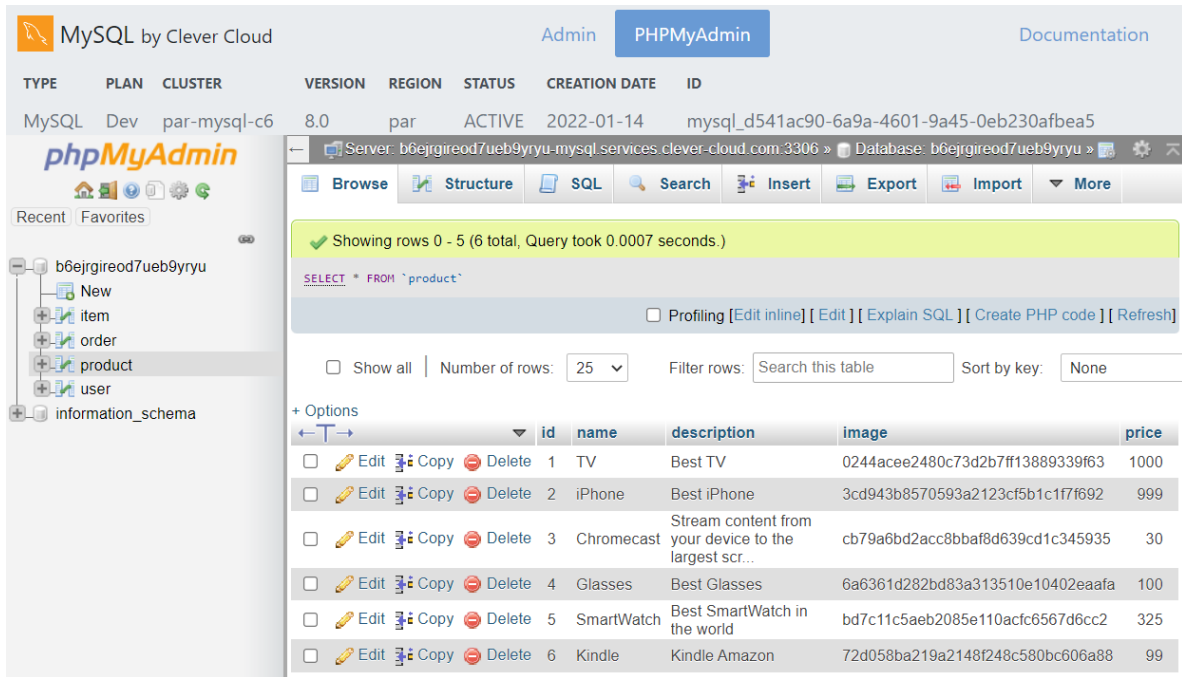


Figure 29-7. Browsing the *product* table.

```
Logged in as books@danielgara.com
Creating app... done, infinite-stream-65085
https://infinite-stream-65085.herokuapp.com/ | https://git.heroku.com/infinite-stream-65085.git
```

Figure 30-1. Creating a Heroku application.

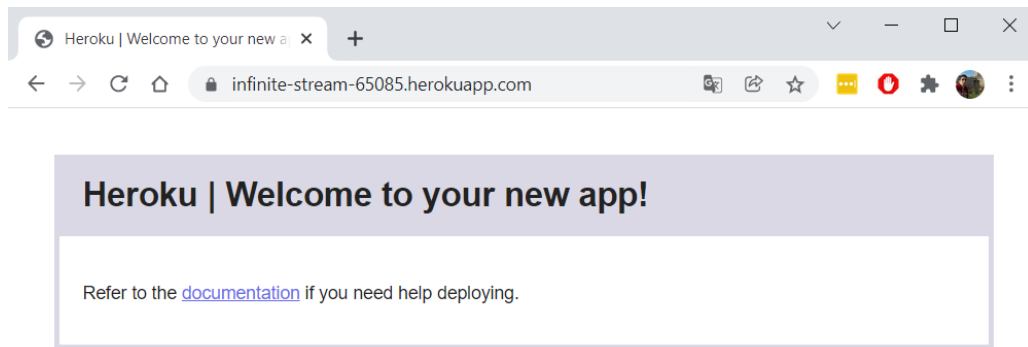


Figure 30-2. Accessing the Heroku application.

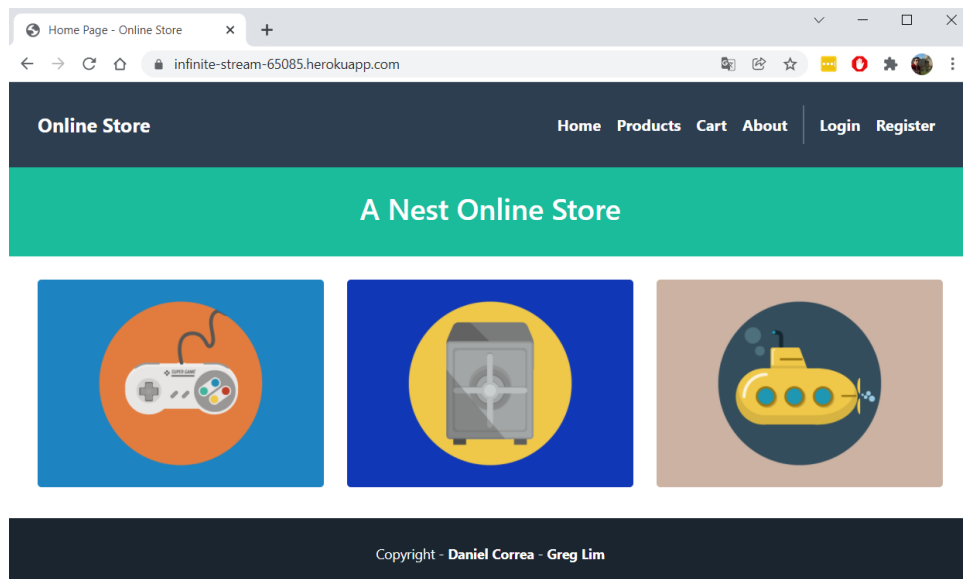


Figure 30-3. Online Store running over the cloud.