IBU | International Burch University

CEN 308 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

Black Steel E-Commerce

Prepared by:
Mahmut Beširević
Asim Veledarević

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

23.06.2022

TABLE OF CONTENTS

1. Introduction **2**
1.1. About the Project 2
1.2. Project Functionalities and Screenshots 2
1.2.1 Functionalities 2
1.2.2 Project Screenshots 3
2. Project Structure **6**
2.1. Technologies 6
2.2. Database Entities 7
2.3. Architectural Pattern 10
2.4. Design Patterns 11
3. Conclusion **11**

1. Introduction

1.1. About the Project

Describe the project/application you were working on in a few sentences and provide *a link* to where it is deployed.

This project resembles an ecommerce platform, specifically tailored for a manufacturer we came up with. It represents a classical web shop with all of the functionalities one might have, while adding a sleek design in the mix.

It was quite a challenge getting all of the backend functionalities to work properly and as intended. We managed to implement a lot of the functionalities, apart from the payments which required using a 3rd party API, due to that we stuck with all of the things we could do ourselves.

The frontend of the application is hosted on heroku, and is available on the following link: https://black-steel.herokuapp.com/.

The backend of the application (the API) is hosted on the following link: https://besirevic.dev/.

1.2. Project Functionalities and Screenshots

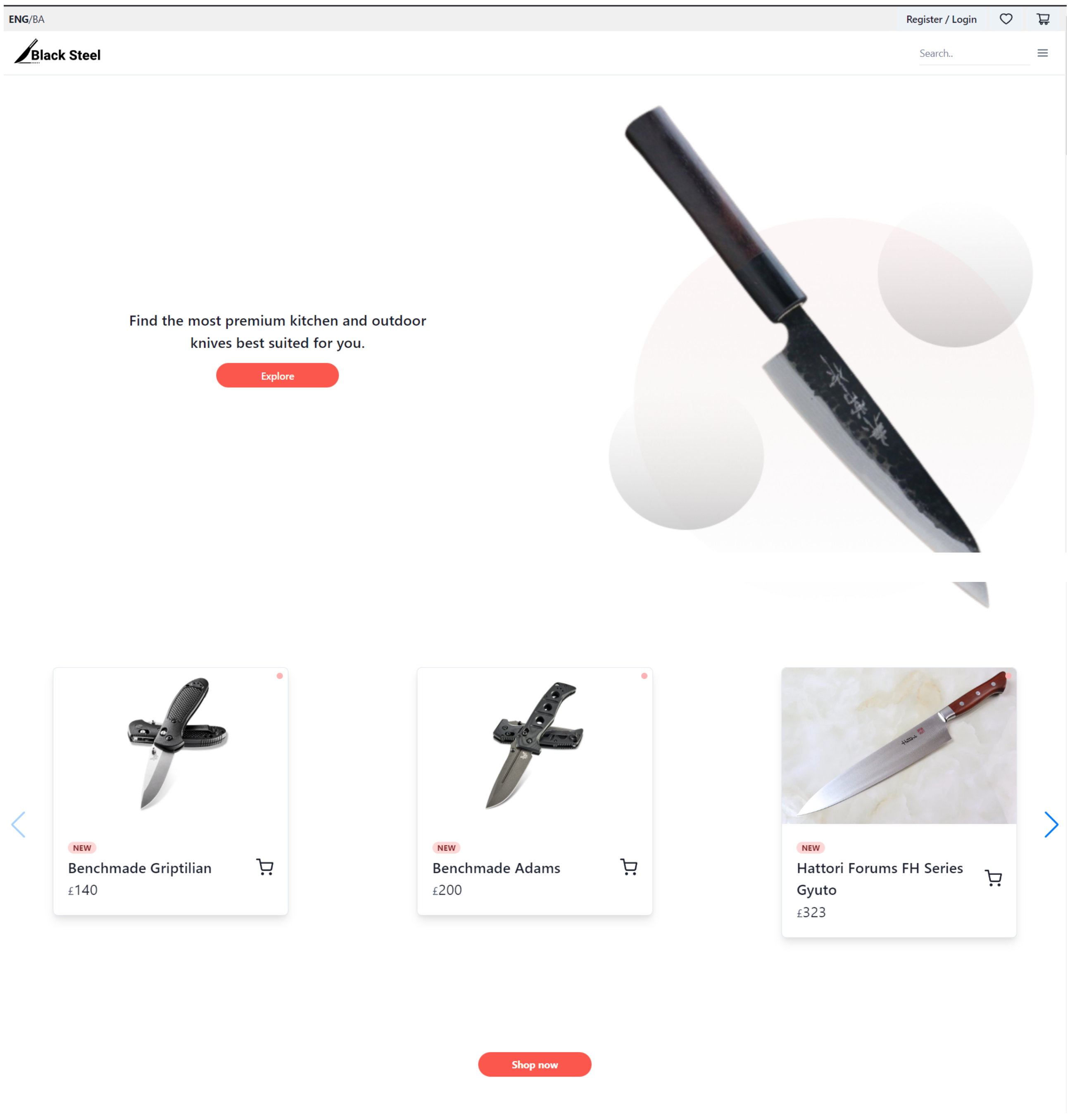
1.2.1 Functionalities

- User Profile
  - Register/login
  - User roles (not implemented in UI)
- Shop
  - View all products
  - Search for specific product
  - Browse by category
- Cart
  - Adding items to cart
  - Viewing your cart
  -
- Checkout
  - Payment and delivery info
- Orders
  - View current orders
  - Order status notifications
- Admin panel

- Adding new products
- Shop management

1.2.2 Project Screenshots

In this section we have placed screenshots that take the regular user flow of the application, as well as showcasing how the website looks and feels.

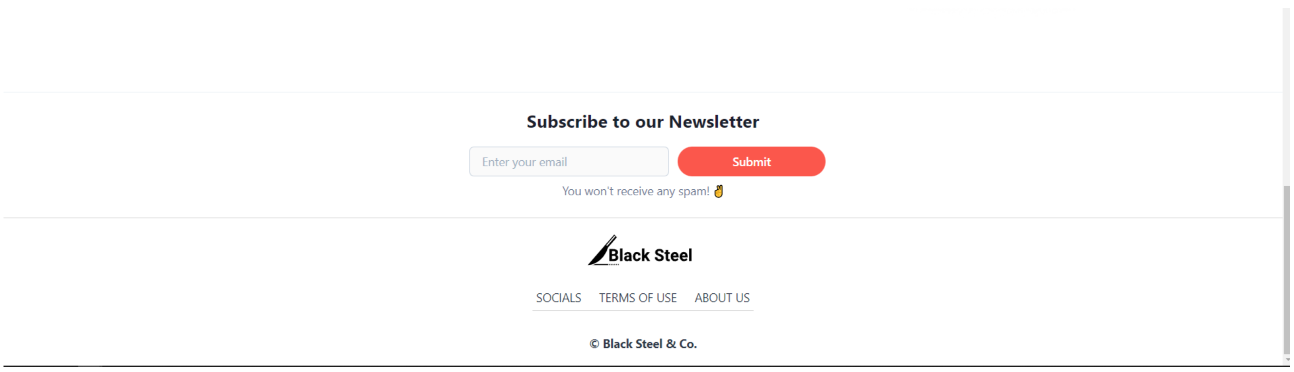


Find the most premium kitchen and outdoor knives best suited for you.

Benchmade Griptilian
£140

Benchmade Adams
£200

Hattori Forums FH Series Gyuto
£323



Subscribe to our Newsletter

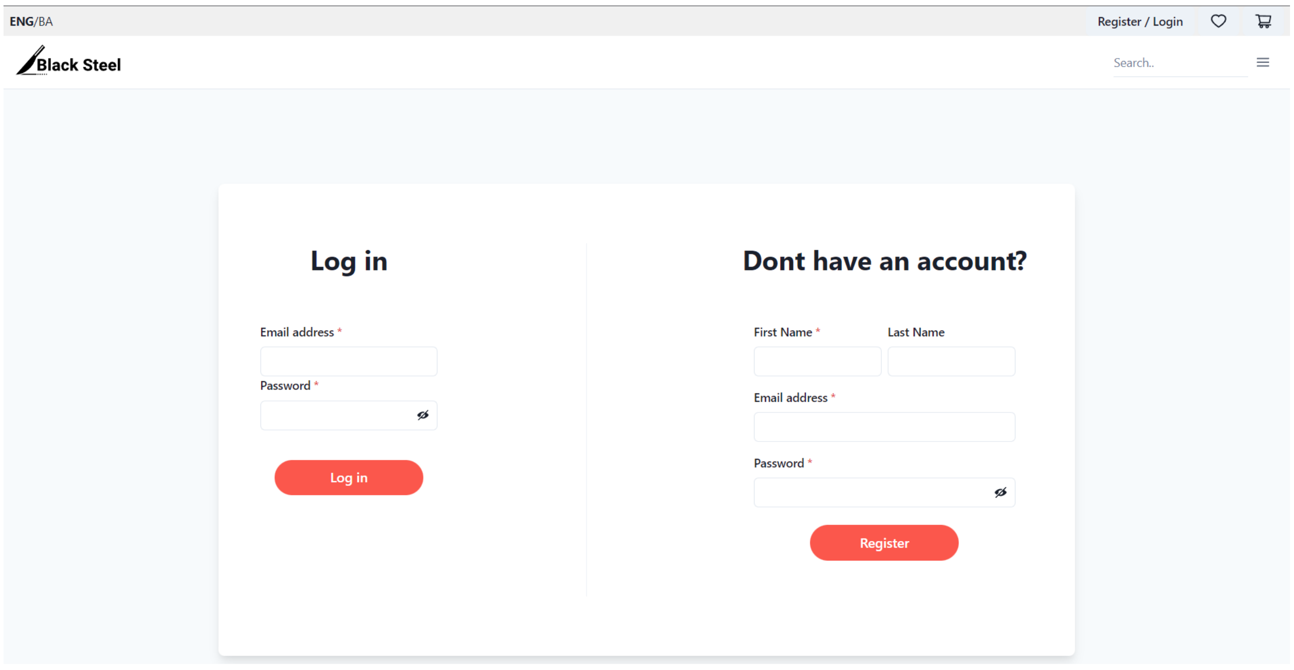
Enter your email | Submit

You won't receive any spam! ✌

Black Steel

SOCIALS TERMS OF USE ABOUT US

© Black Steel & Co.



ENG/BA Register / Login

Black Steel Search..

Log in

Email address *

Password *

Log in

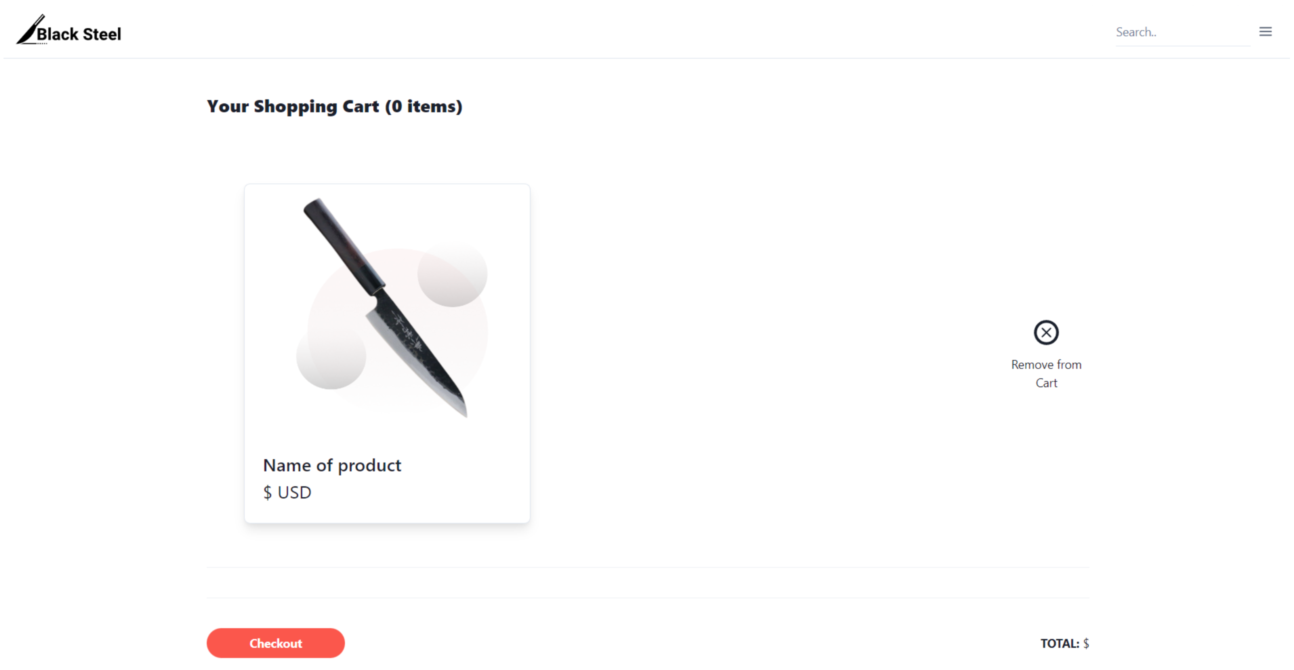
Dont have an account?

First Name * Last Name

Email address *

Password *

Register



Black Steel Search..

Your Shopping Cart (0 items)

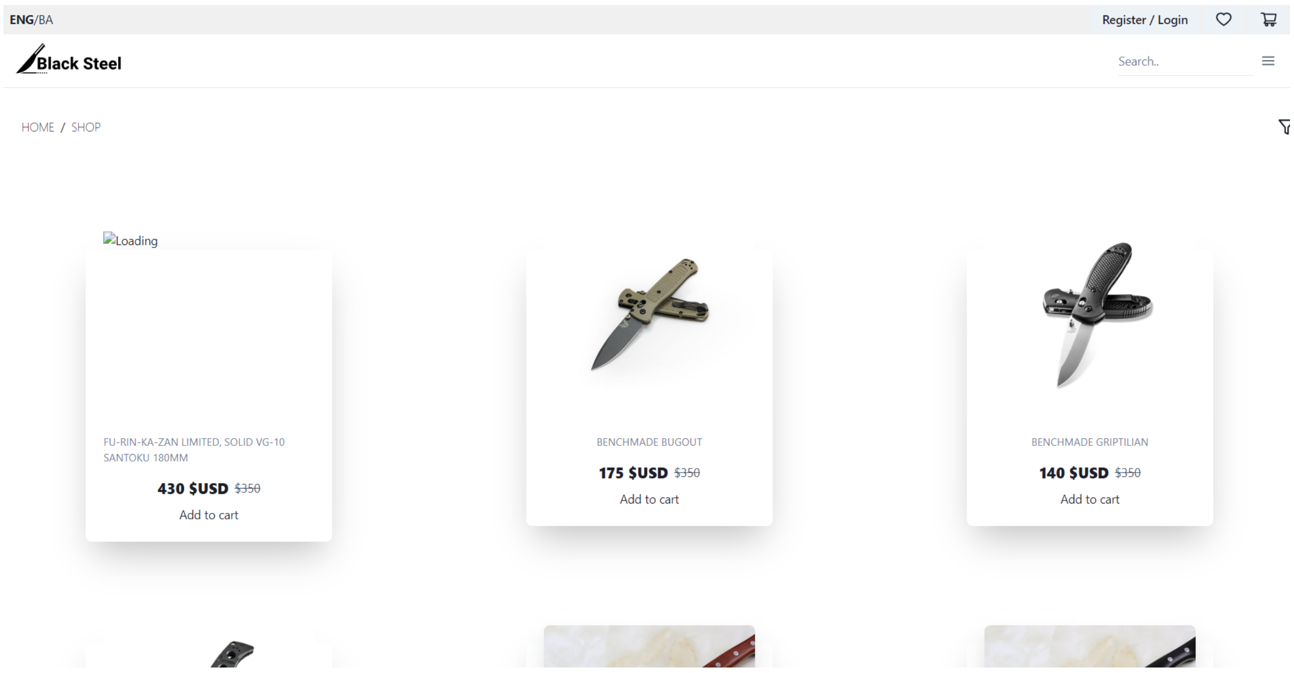
Name of product

$ USD

Remove from Cart

Checkout

TOTAL: $


ENG/BA
Register / Login

Black Steel

Search..

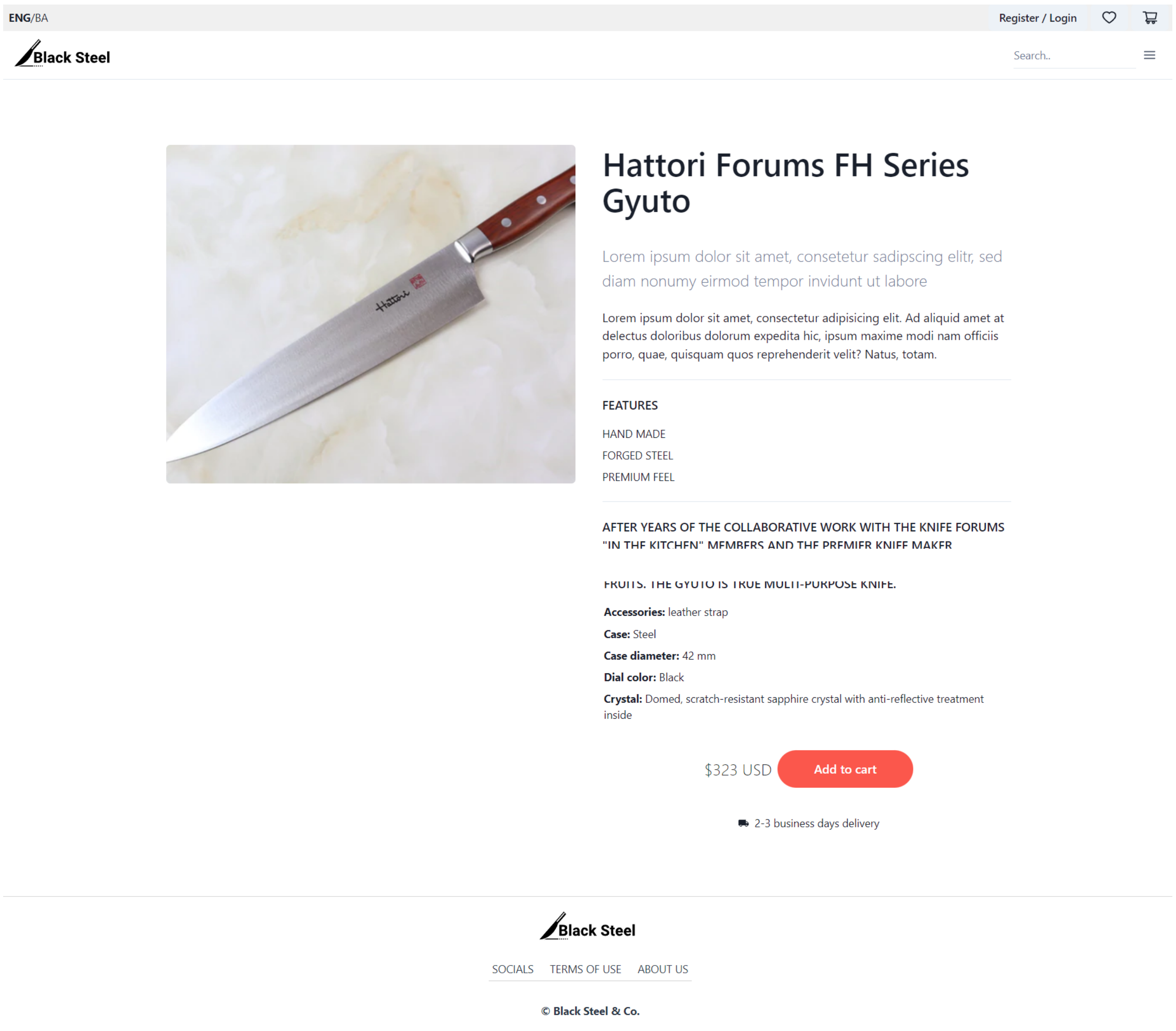
HOME / SHOP

Loading

FU-RIN-KA-ZAN LIMITED, SOLID VG-10 SANTOKU 180MM
430 $USD ~~$350~~
Add to cart

BENCHMADE BUGOUT
175 $USD ~~$350~~
Add to cart

BENCHMADE GRIPTILIAN
140 $USD ~~$350~~
Add to cart


ENG/BA
Register / Login

Black Steel

Search..

Hattori Forums FH Series Gyuto

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad aliquid amet at delectus doloribus dolorum expedita hic, ipsum maxime modi nam officiis porro, quae, quisquam quos reprehenderit velit? Natus, totam.

FEATURES

HAND MADE
FORGED STEEL
PREMIUM FEEL

AFTER YEARS OF THE COLLABORATIVE WORK WITH THE KNIFE FORUMS "IN THE KITCHEN" MEMBERS AND THE PREMIER KNIFE MAKER

FRUITS. THE GYUTO IS TRUE MULTI-PURPOSE KNIFE.

Accessories: leather strap
Case: Steel
Case diameter: 42 mm
Dial color: Black
Crystal: Domed, scratch-resistant sapphire crystal with anti-reflective treatment inside

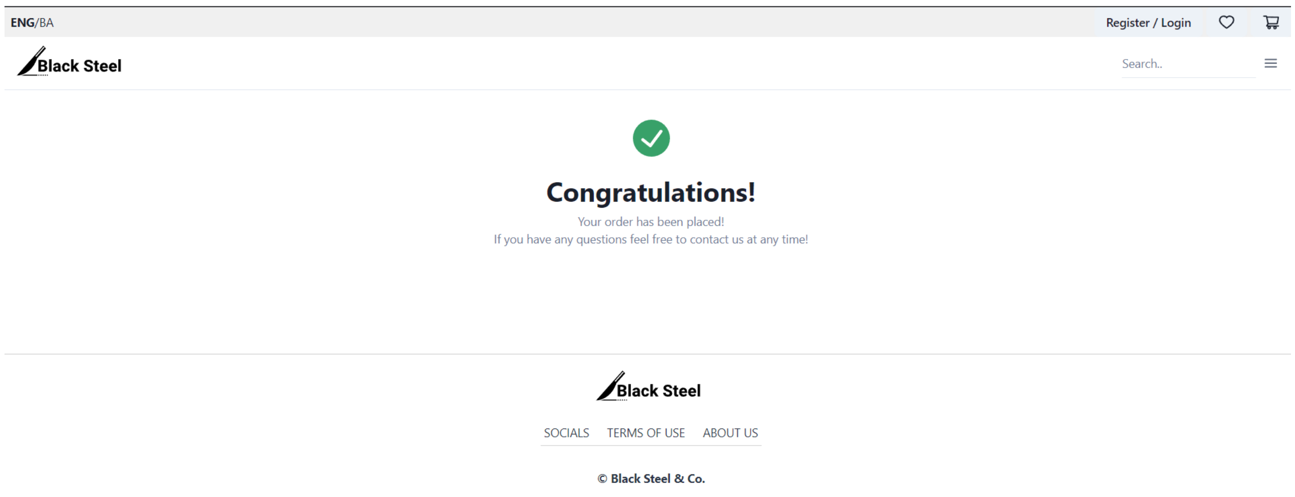
$323 USD Add to cart

2-3 business days delivery

Black Steel

SOCIALS TERMS OF USE ABOUT US

© Black Steel & Co.


ENG/BA

Register / Login

Black Steel

Search..

Congratulations!

Your order has been placed!
If you have any questions feel free to contact us at any time!

Black Steel

SOCIALS TERMS OF USE ABOUT US

© Black Steel & Co.

2. Project Structure

2.1. Technologies

Here we provide a short overview of all of the technologies used to create the project:

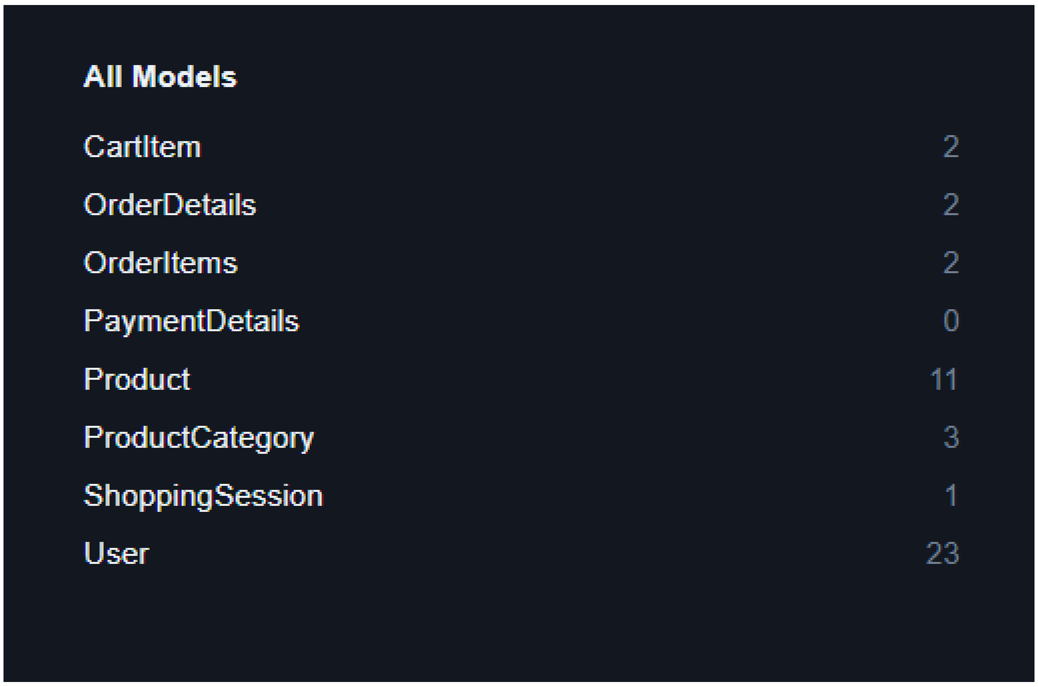
- Hosting : VPS with nginx; cloudflare, heroku
- CI/CD : Jenkins
- Database : PostgreSQL
- Backend :
  - Language : JavaScript
  - Frameworks : NodeJS, ExpressJS, PrismaJS(ORM)
- Frontend:
  - Language : JavaScript
  - Frameworks : NextJS (React)
  - Styling: Chakra UI

We opted to use the ES6 coding standard, and as a reference used the AirBnB coding style guide, to help us better understand and notice where to use it.

The style guide can be found on the following link: https://github.com/airbnb/javascript.

2.2. Database Entities

Below we have listed all of the database columns with their respective properties, we note that the full database schema can also be found in the black-steel-api repository under /prisma/schema.prisma.



All Models

CartItem	2
OrderDetails	2
OrderItems	2
PaymentDetails	0
Product	11
ProductCategory	3
ShoppingSession	1
User	23

- user
    - id Int
    - createdAt DateTime
    - email String
    - name String
    - surname String
    - password String
    - address String?
    - role Role
    - Cart ShoppingSession?
    - OrderDetails OrderDetails[]
- product
    - id Int
    - name String

    - description String
    - price   Float
    - brand   String
    - category_id Int?
    - categories ProductCategory? @relation(fields: [category_id],references: [id])
    - no_in_stock Int
    - img   String
    - date_added DateTime
    - order_item OrderItems?
    - CartItem   CartItem?
- product_category
    - id   Int
    - name   String
    - desc   String
    - created_at DateTime
    - Product   Product[]
    -

- cart
    - id   Int
    - user_id   Int
    - totalPrice Decimal
    - createdAt DateTime
    - modifiedAt DateTime
    - user   User
    - CartItem   CartItem[]
    -
- cart_item
    - id   Int
    - session_id   Int
    - product_id   Int
    - quantity   Int

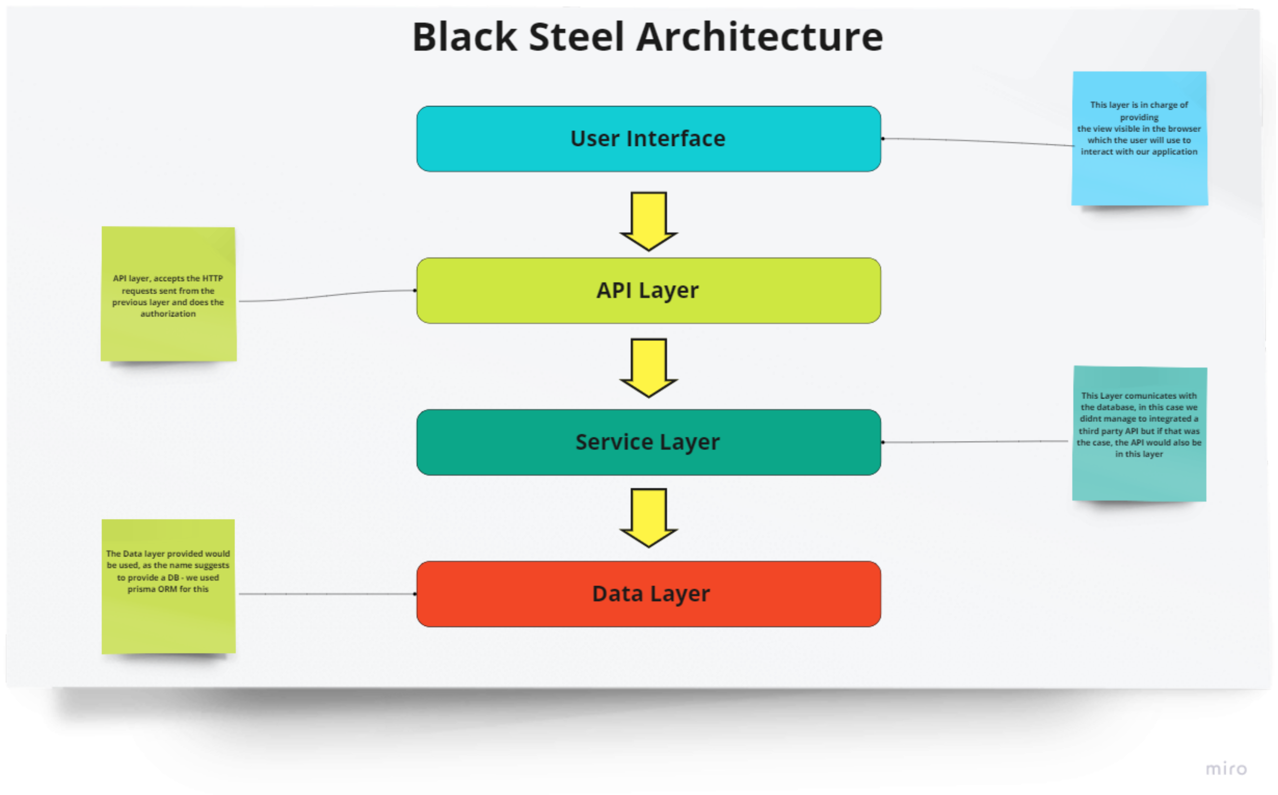
    - createdAt DateTime
    - ShoppingSession ShoppingSession @relation(fields: [session_id], references: [id])
    - Product Product @relation(fields: [product_id], references: [id])
- order_details
    - id Int
    - user_id Int
    - total Decimal
    - createdAt DateTime
    - updatedAt DateTime?
    - payment_confirmed Boolean
    - OrderItems OrderItems[]
    - payment_id PaymentDetails?
    - user User @relation(fields: [user_id],references: [id])
    -
- order_items
    - id Int
    - order_id Int
    - product_id Int
    - quantity Int
    - createdAt DateTime
    - product Product @relation(fields: [product_id],references: [id])
    - order_details OrderDetails @relation(fields: [order_id], references: [id])

- OrderStatus
    - id Int
    - user_id Int
    - order_id Int
    - delivery_status String
    - shipment_method String
    - delivery_notes String?
    - delivered Boolean
    -

- payment_details
    - id Int
    - order_id Int
    - amount Int
    - provider String
    - status String
    - currency String
    - cc_number String
    - cc_name String
    - cc_exp_date DateTime
    - cc_display_number String
    - cc_company String
    - country String
    - createdAt DateTime
    - modifiedAt DateTime
    - OrderDetails OrderDetails @relation(fields: [order_id],references: [id])

2.3. Architectural Pattern

The architectural pattern used is the **Layered pattern**. It was chosen since we need to be able to serve the clients via the Presentation layer, while processing information and changing it in the Data layer via the Business layer.


Black Steel Architecture
This layer is in charge of providing the view visible in the browser which the user will use to interact with our application
User Interface
API layer, accepts the HTTP requests sent from the previous layer and does the authorization
API Layer
This Layer comunicates with the database, in this case we didnt manage to integrated a third party API but if that was the case, the API would also be in this layer
Service Layer
The Data layer provided would be used, as the name suggests to provide a DB - we used prisma ORM for this
Data Layer
miro

Explaining the layers and our thought process:

- User Interface - This layer is in charge of providing the view visible in the browser which the user will use to interact with our application

- API Layer - API layer that accepts the HTTP requests sent from the previous layer and does the authorization

- Service Layer - This Layer communicates with the database, in this case we didn't manage to integrated a third party API but if that was the case, the API would also be in this layer

- Data layer - The Data layer provided would be used, as the name suggests to provide a DB - we used prisma ORM for this

2.4. Design Patterns

- **Composite**
  - Use: The way our database is structured and connected, we used the composite pattern to easily calculate the total price of orders
- **Builder**
  - Use: For merging the cart and cart items into a single order

3. Conclusion

The overall project is implemented with working backend and frontend, both being hosted on heroku(frontend) and a VPS(backend). The implementation of the project was a bit tedious but we managed to implement most of the things and make it work as intended.

A couple of difficulties did occur but with the combined efforts we managed to overcome many of them. The hardest part was handling the authorisation and sessions for the cart and orders.