

**E-commerce**

**(FINAL PROJECT REPORT)**

By

**Dwipkumar Patel**

Student ID : 2014100

Program Name

**AEC INFORMATION TECHNOLOGY PROGRAMMER-ANALYST – LEA.3Q**

Project Supervisor at the College

**Dara Aghamirkarimi**

**2021**

Table of Contents

[**Acknowledgement** 3](#_Toc74588436)

[Context: 4](#_Toc74588437)

[Why are you doing the Project? 4](#_Toc74588438)

[What are your learning expectations/goals? 4](#_Toc74588439)

[About Project With diagram. 5](#_Toc74588440)

[Technology used. 8](#_Toc74588441)

[Dependencies that we have used: 9](#_Toc74588442)

[DataBase 11](#_Toc74588443)

[User Inter Face 13](#_Toc74588444)

[User Interface Admin Side 19](#_Toc74588445)

[PostMan Screenshot 24](#_Toc74588446)

[Conclusion 24](#_Toc74588447)

[References 25](#_Toc74588448)

# **Acknowledgement**

Learning new technology is a big challenge but when you take the challenge as your goal, without fail you will achieve it and SUCCESS is certain. For this reason, we would like to express our deepest gratitude to all those involved in making this task a success. This project would have not been completed successfully without the help of our community.

Before all else, we would like to thank **Professor Dara Aghamirkarimi**, who is a very challenging guru who also challenges student’s ability and make them think beyond the horizon. Additionally, with his encouragement and guidance, we got a chance to explore and learn new technologies and skillsets.

Not missing to thank our own Department of Information Technology, for giving us this opportunity to gain practical experience and enable us to understand the demand of the experience and professional environment. I have gained valuable experience which has given confidence and has enhanced professional skill for the competent career.

We also would like to thank our family member and friends for helping, bearing, encouraging, and boosting moral throughout our project period. Finally, we would like to express sincere gratitude to our developer’s community for helping us directly or indirectly to complete our project.

# Context:

This project on E commerce in which user can register to buy product from the site where with as many quantities user want buy user can write reviews and give them star from 5, from admin side, admin can manage user, products, and order where he can update or delete. It is MERN stack application, which is full dynamic working, with MongoDB.

## Why are you doing the Project?

This is a e commerce web site in which have use react for my frontend and backend, MangoDB for data base.

In this project user can create account and buy product with the quantity and also write a review of the product and give them rating out of 5.

For payment I have use Pay Pal API, so we can use pay pal account along with that we can use credit and debit card.

From the admin side the web site can be controlled like admin can manage user and he is also able to delete the user or update it.

Admin can add product with some details and image, he can also mange order which has been created by user and after conformation he can mark as delivered.

## What are your learning expectations/goals?

My expectations and goals are:

As, I want to work in web development industry.

To have logical experience I have develop this web site in which I can update, delete, insert in one page application.

Plan, design and build my own Information System, useful, profitable and easy to use.

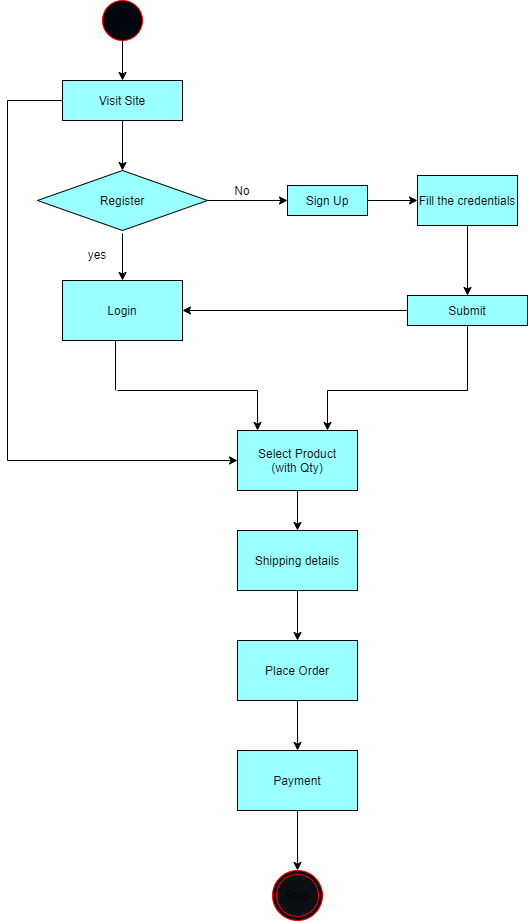
As from this I have make my had strong in react (JavaScript) and as well in node.js.

For backend I have make grip in postman and MangoDB.

And most importantly I have learned about time limit and complete my project in time.

I will be updating my sit as per requirement in future.

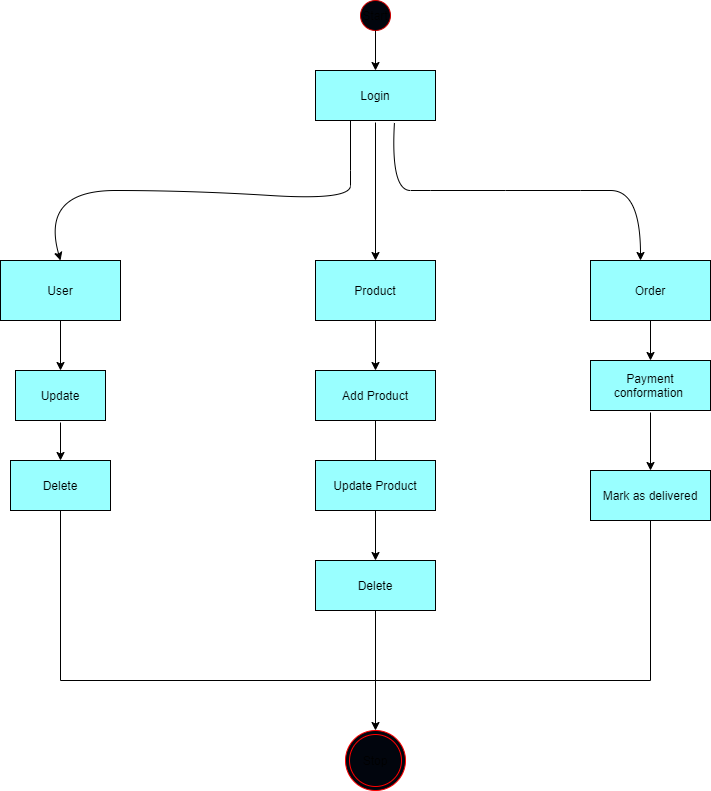
# About Project With diagram.



The above flow chart can show the functionality of the user side

When user enter in site they can register account with some basic information or if they already register they can directly login in the website, after that they can go site to buy the product along with that we can read the reviews of the previous customer in product details description.

After selecting the with the Quantity we can follow-up to purchase order, after providing shipping details we can go to payment side in which we have do use PayPal, or Credit/Debit card ,at next step we can place order.



In the above diagram show that how admin part works,

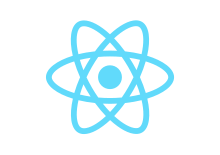
User inter face is same for admin and customer while it differentiate after login admin has option in navigation bar to manage product, orders and users.

In user section admin has rights to remove user or make as admin , also has ability to check details of user.

In Order section admin can see about order and theirs details along with that admin can make order has deliver to customer with date and time.

In product section admin can add product with details and images, admin can also change previous order details, they can remove the product also.

# Technology used.

***React.js***: It facilitates fast and interactive user interfaces for web and mobile applications. It’s an open-source library with a component-based approach. In a Model View Controller architecture, react is the View layer, which is responsible for how the app looks and feels.

***Node.js***: A server-side programming framework**.** It isbuilt on Chrome’s JavaScript engine uses asynchronous programming and this is the main reason for using it

***Express.js***: It allows you to create a REST API server. Easy to connect with databases such as MongoDB. Easy to configure and customize. It allows you to define routes of your application based on HTTP methods and URLs.

***MongoDB***: Has Document-Oriented storage. Has a single master with built-in replication support. Ships with built-in sharing support ( *spreading data around several replicated clusters*). It has very wide driver support, although databases ship with limited support, MongoDB has drivers for every language. Provide horizontal scalability to balance the increasing load of modern applications. Robust, flexible, and scale able.

***MongoDB Compass*** is the GUI for MongoDB. Compass allows you to analyze and understand the contents of your data without formal knowledge of MongoDB query syntax. In addition to exploring your data in a visual environment, you can also use Compass to optimize query performance, manage indexes, and implement document validation.

***Redux*** is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to (and inspired by) Facebook's Flux architecture, it was created by Dan Abramov and Andrew Clark.

***Postman*** is an application for testing APIs, by sending request to the web server and getting the response back. It allows users to set up all the headers and cookies the API expects and checks the response. Productivity can be increased using some of the Postman features.

## Dependencies that we have used:

**REACT** (React, n.d.)

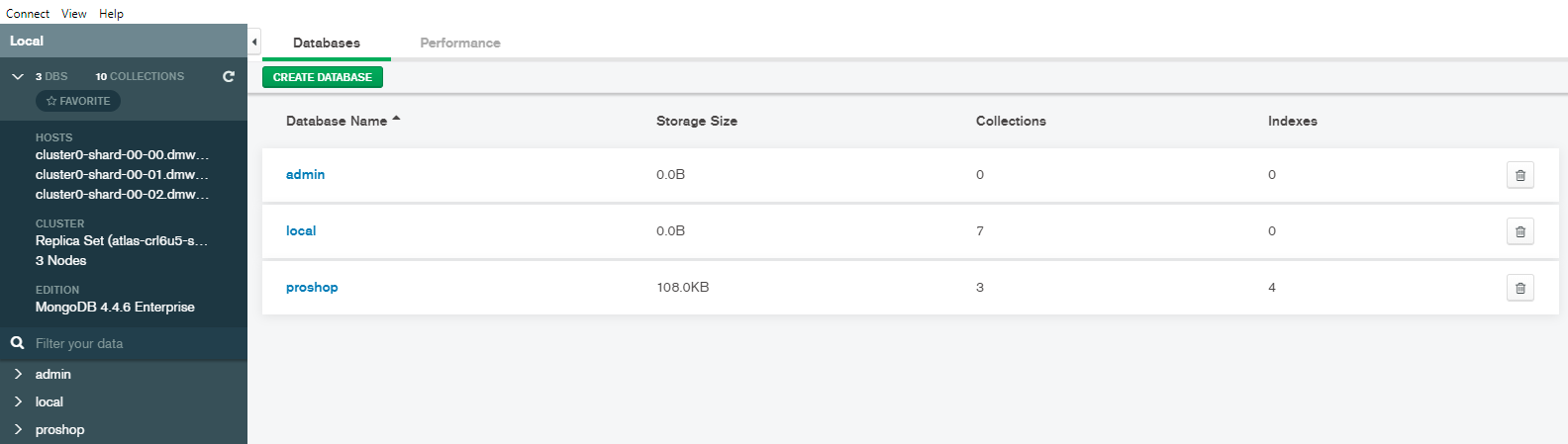
* ***Axios***is a JavaScript library used to make HTTP requests from node.js or XMLHttpRequests from the browser that also supports the ES6 Promise API. (Axios, n.d.)
* ***Moment*** open-source JavaScript library that removes the need to use the native JavaScript Date object directly. (moment, n.d.)
* ***React-file-base-64***React Component for Converting Files to base64. It's based on Dev Mozilla Website. (react-file-base64, n.d.)
* ***Redux*** is a pattern and library for managing and updating application state, using events called "actions". (Redux, n.d.)
* ***Redux Thunk*** is a middleware that lets you call action creators that return a function instead of an action object. That function receives the store’s dispatch method, which is then used to dispatch regular synchronous actions inside the function’s body once the asynchronous operations have been completed. (Redux-thunk, n.d.)
* ***Material-UI* components** work in isolation. They are self-supporting and will only inject the styles they need to display. (Material-UI/core, n.d.)
* ***React-Store***A store holds the whole state tree of your application. The only way to change the state inside it is to dispatch an action on it.

**NODE** (Node, n.d.)

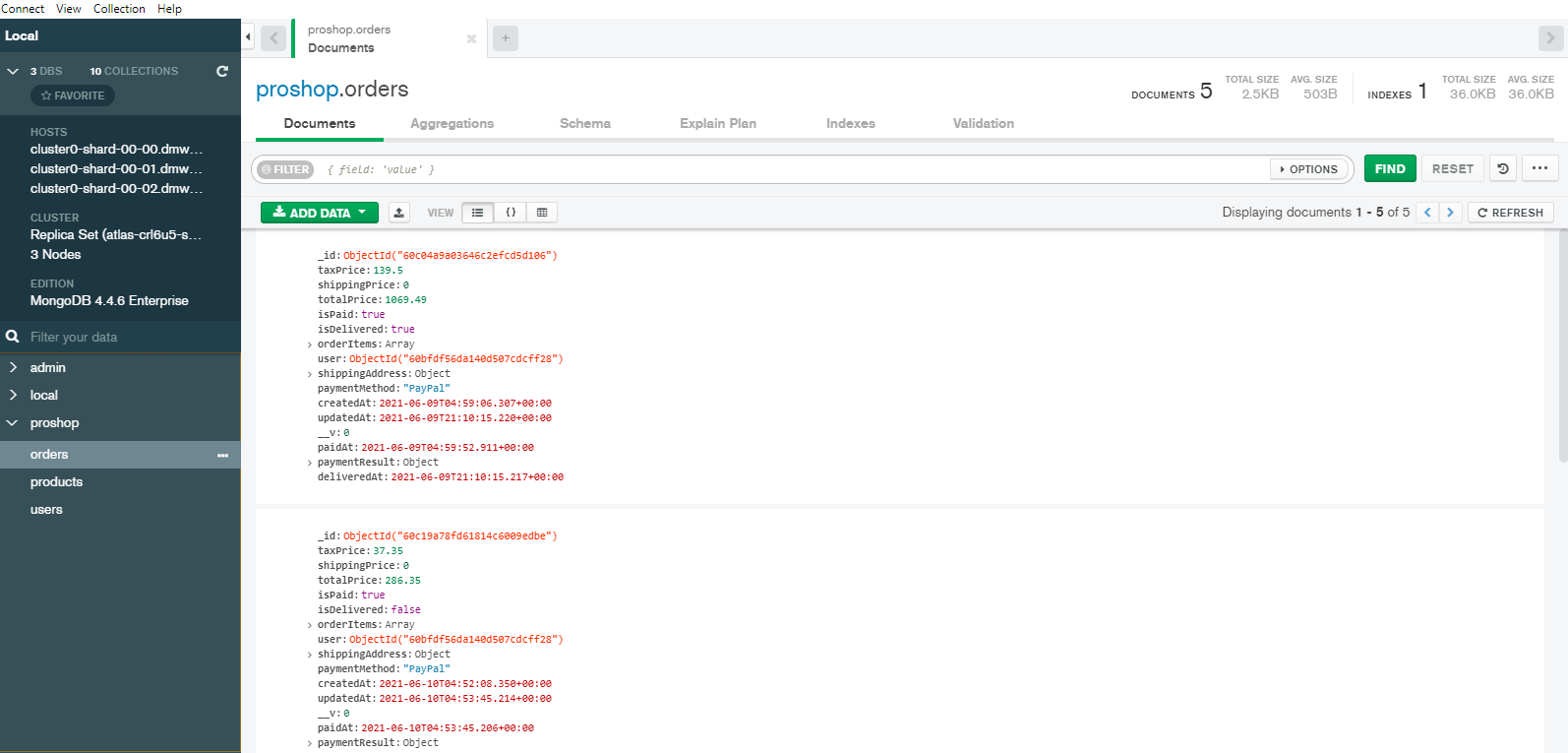
* ***Body-parser*** object exposes various factories to create middlewares. All middlewares will populate the req.body property with the parsed body when the Content-Type request header matches the type of option, or an empty object ({}) if there was no body to parse, the Content-Type was not matched, or an error occurred. (Body-parser, n.d.)
* ***CORS*** *(Cross-Origin Resource Sharing)*. It is a mechanism to allow or restrict requested resources on a web server depend on where the HTTP request was initiated. (Cors, n.d.)
* ***Express***web application server framework, which is specifically designed for building single-page, multi-page, and hybrid web applications. It has become the standard server framework for node. (Express, n.d.)
* ***Mangoose***is an Object Data Modeling (ODM) library for MongoDB and Node. js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB. (Mongoose, n.d.)
* ***Nodemon*** is a tool that helps develop node. js based applications by automatically restarting the node application when file changes in the directory are detected. ... nodemon is a replacement wrapper for node. To use nodemon, replace the word node on the command line when executing your script. (Nodemon, n.d.)

# DataBase

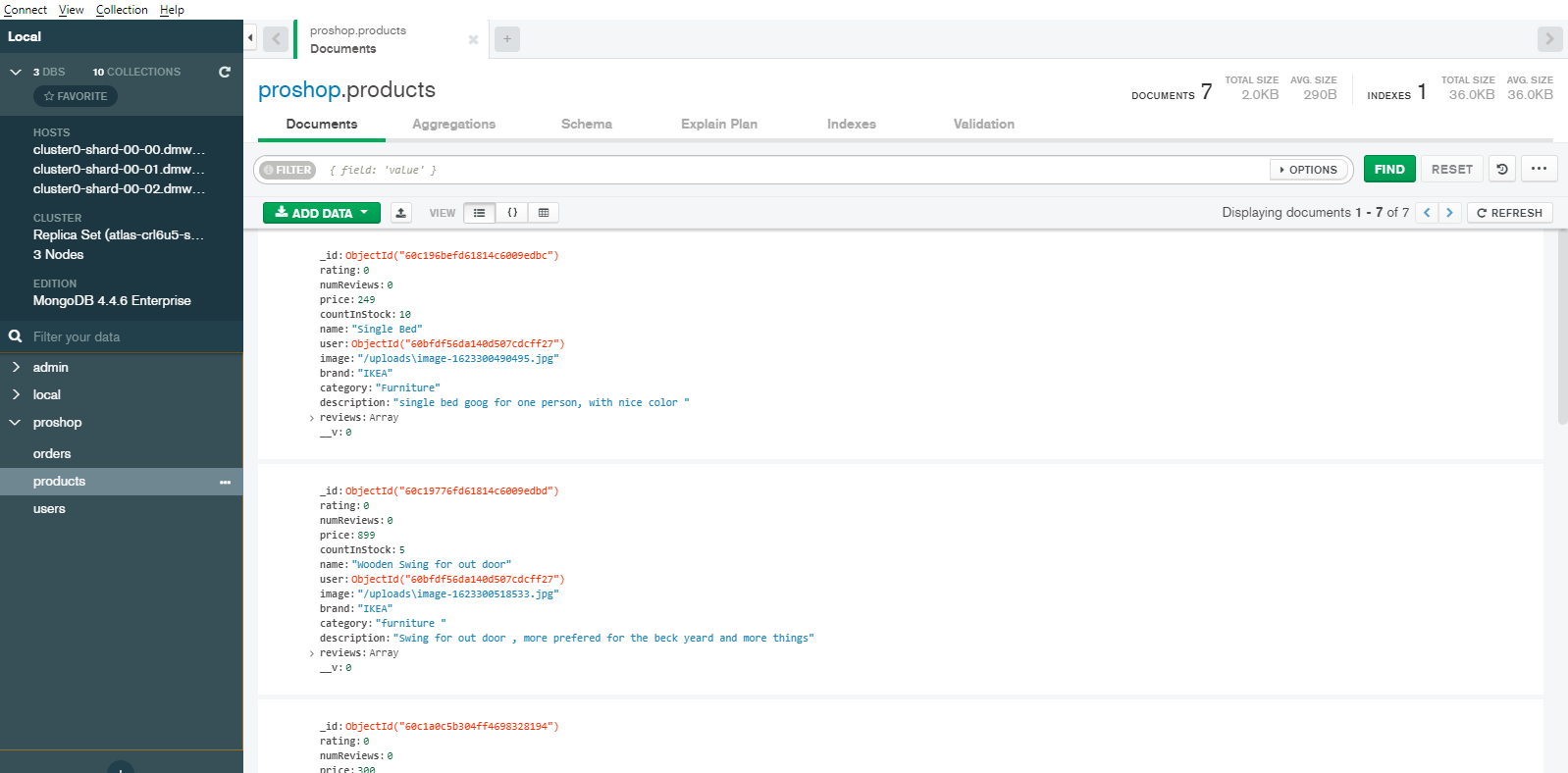
**Table of web site in MangoDB**

****

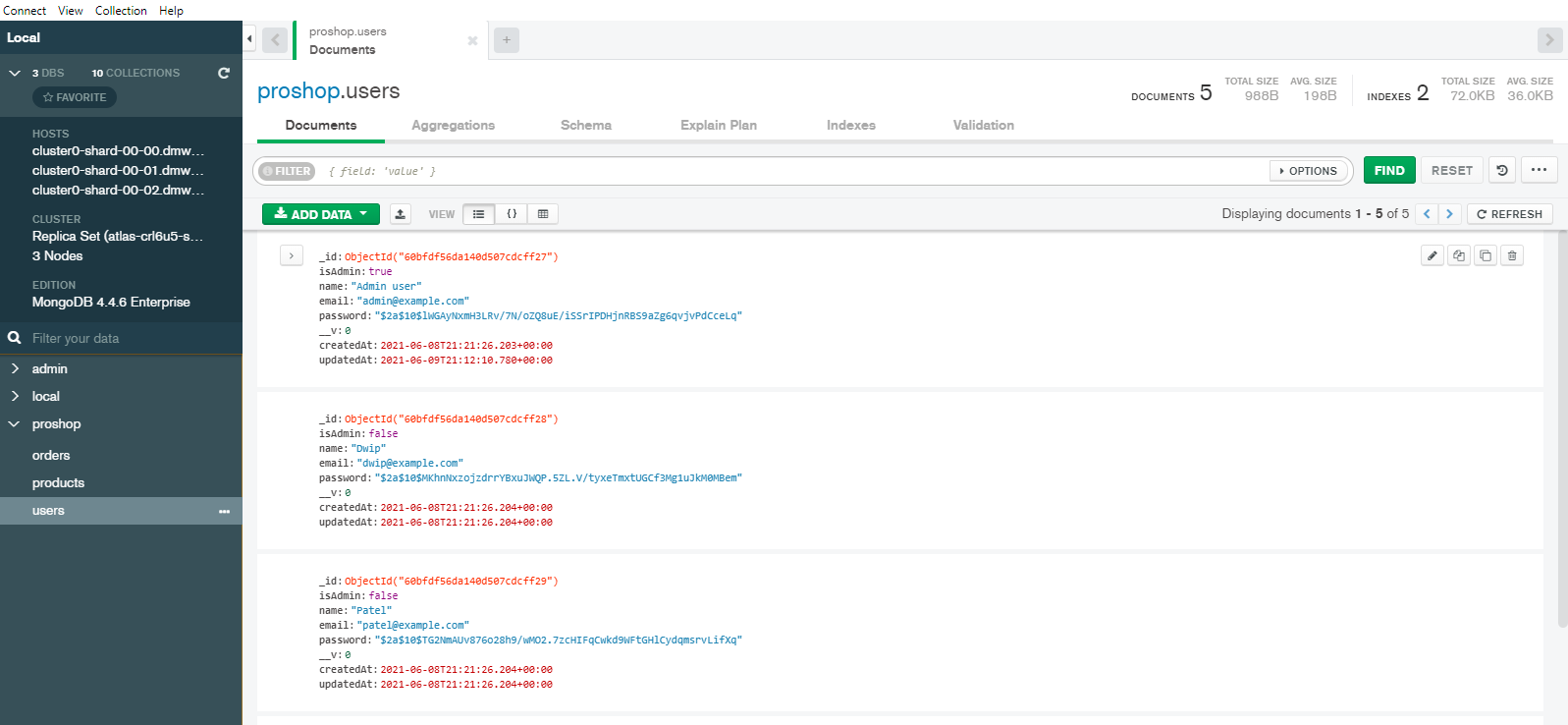
**Table for order in MangoDB**

****

**Table for Product in MangoDB**

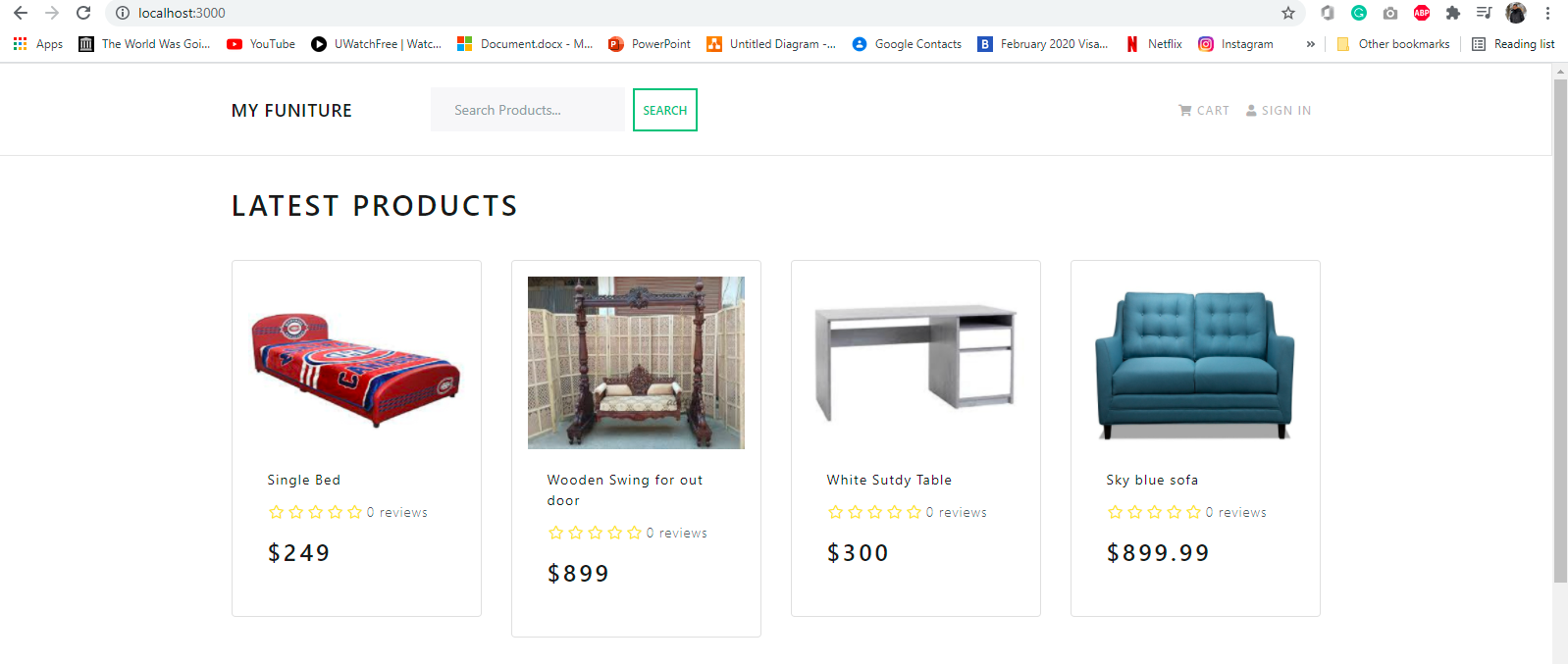
****

**Table for User in MangoDB**

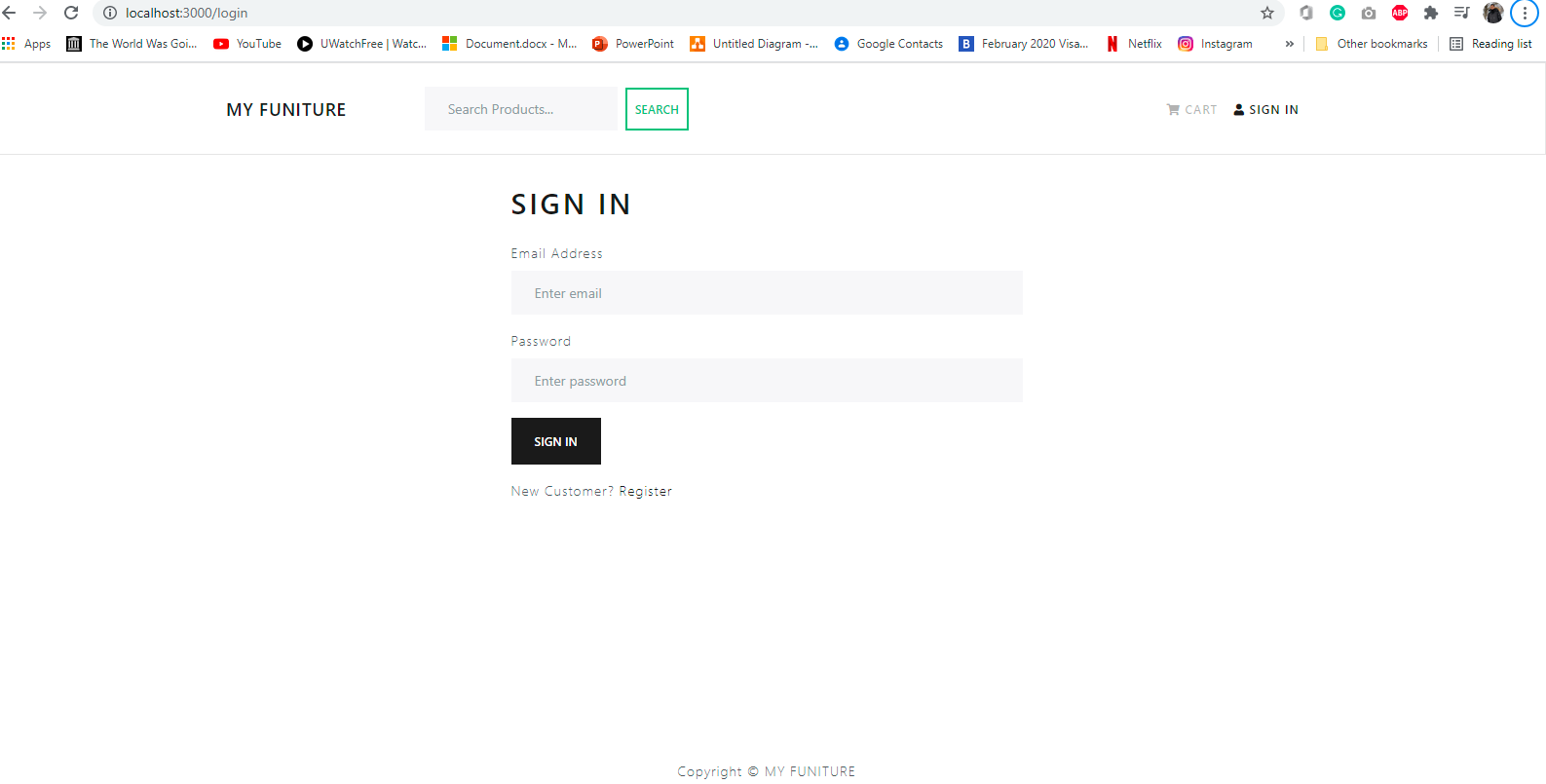
****

# User Inter Face

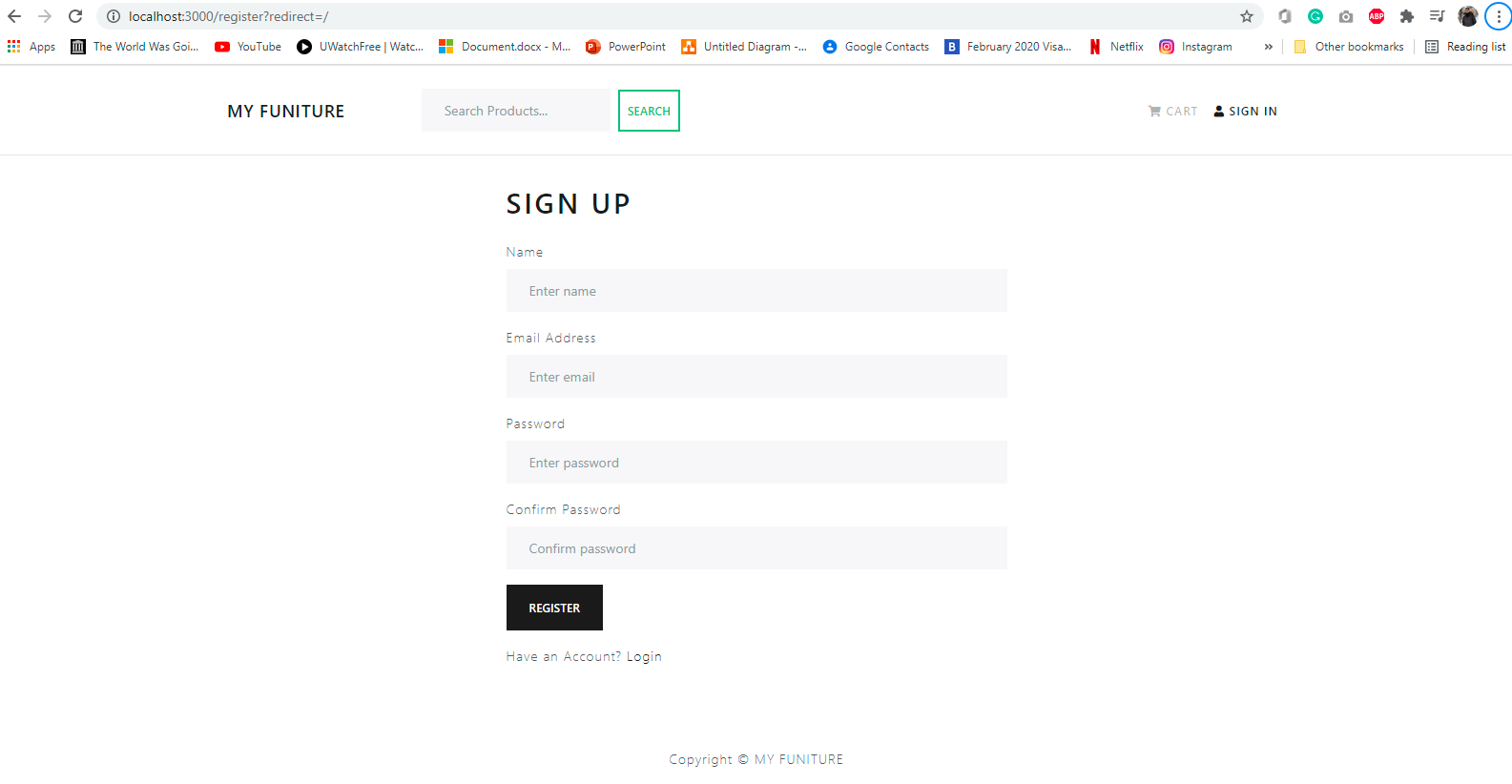
**Project Output screen (Screen Short)**

****

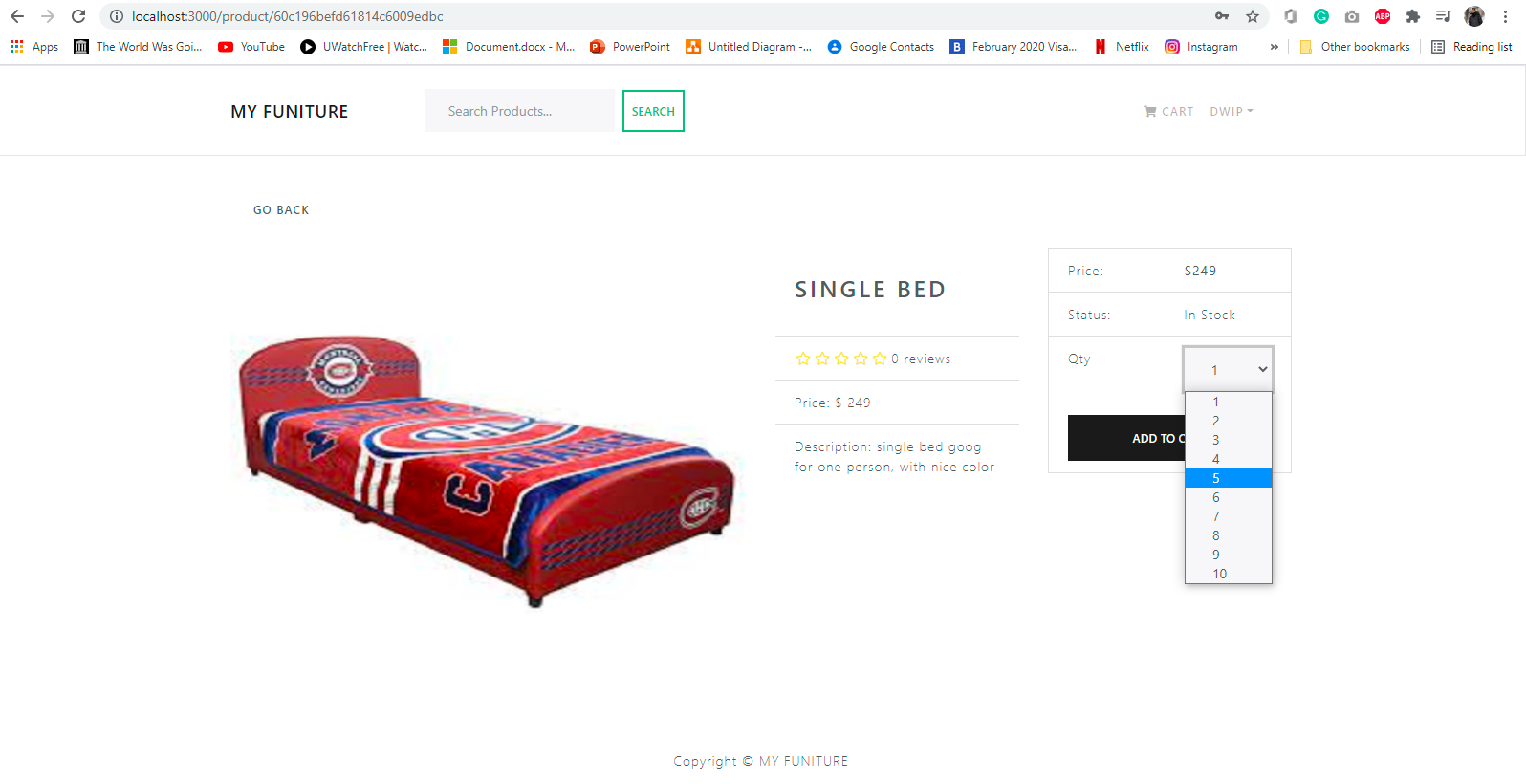
**Login Page**

****

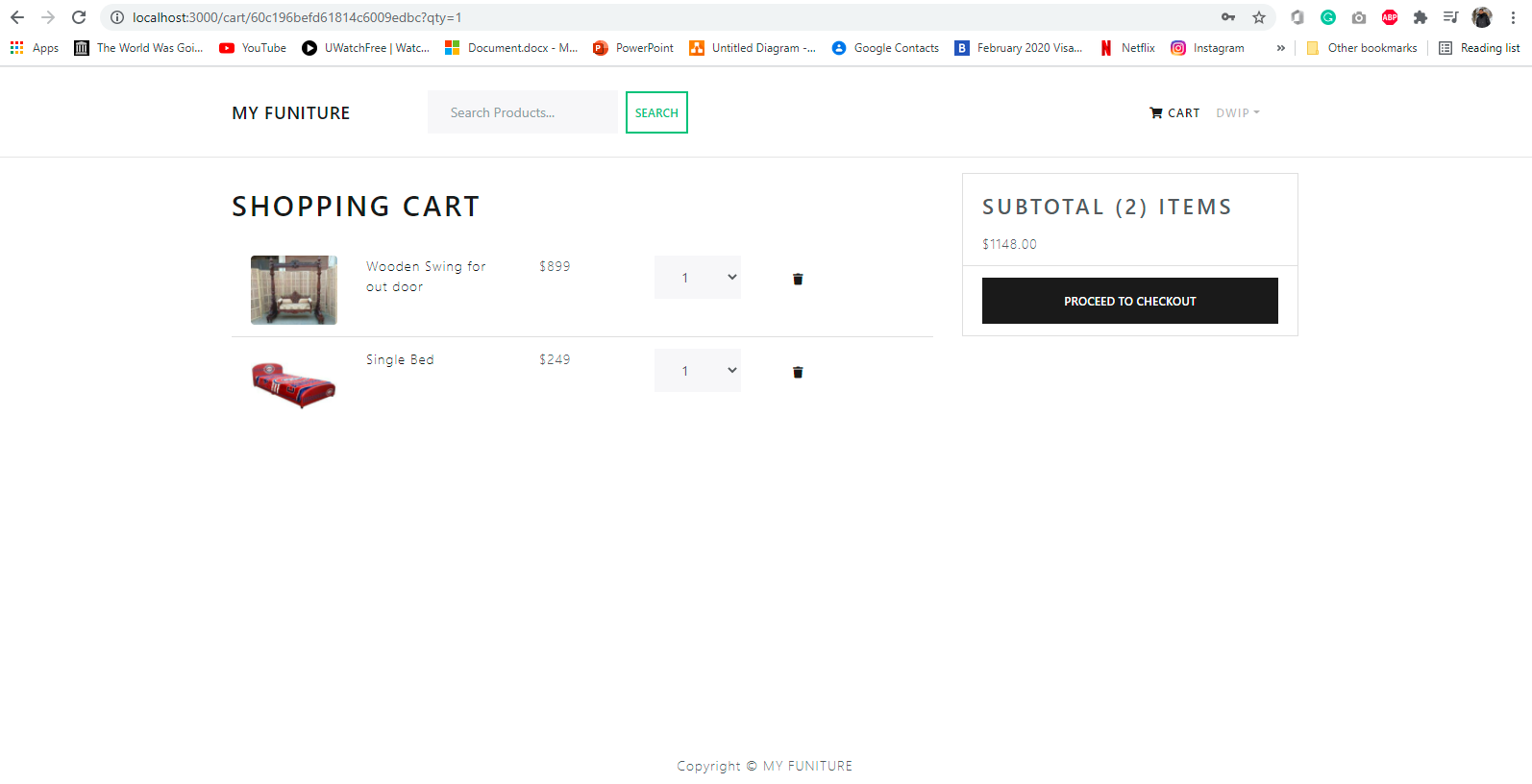
**Register Form**

****

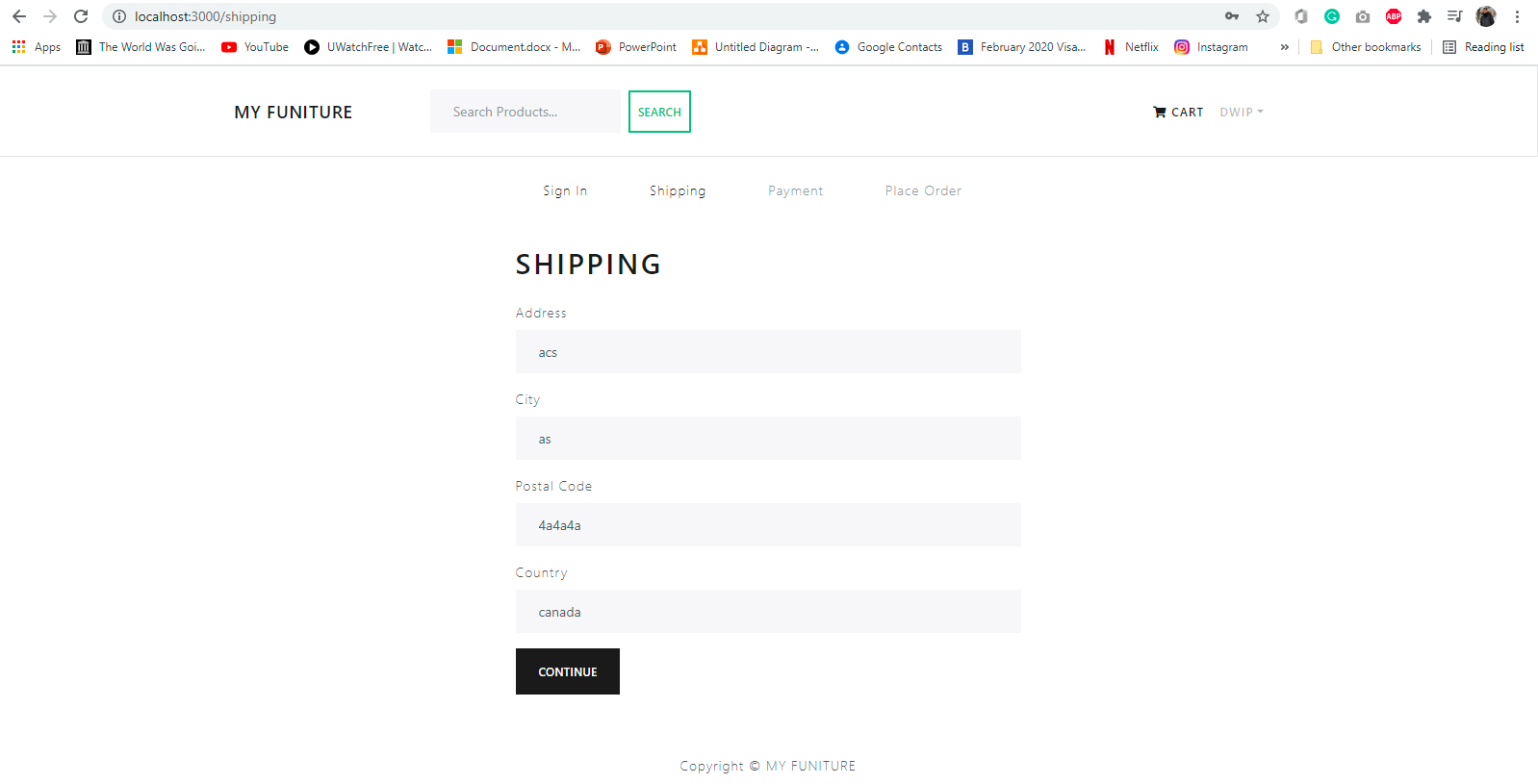
**Product details Page**

****

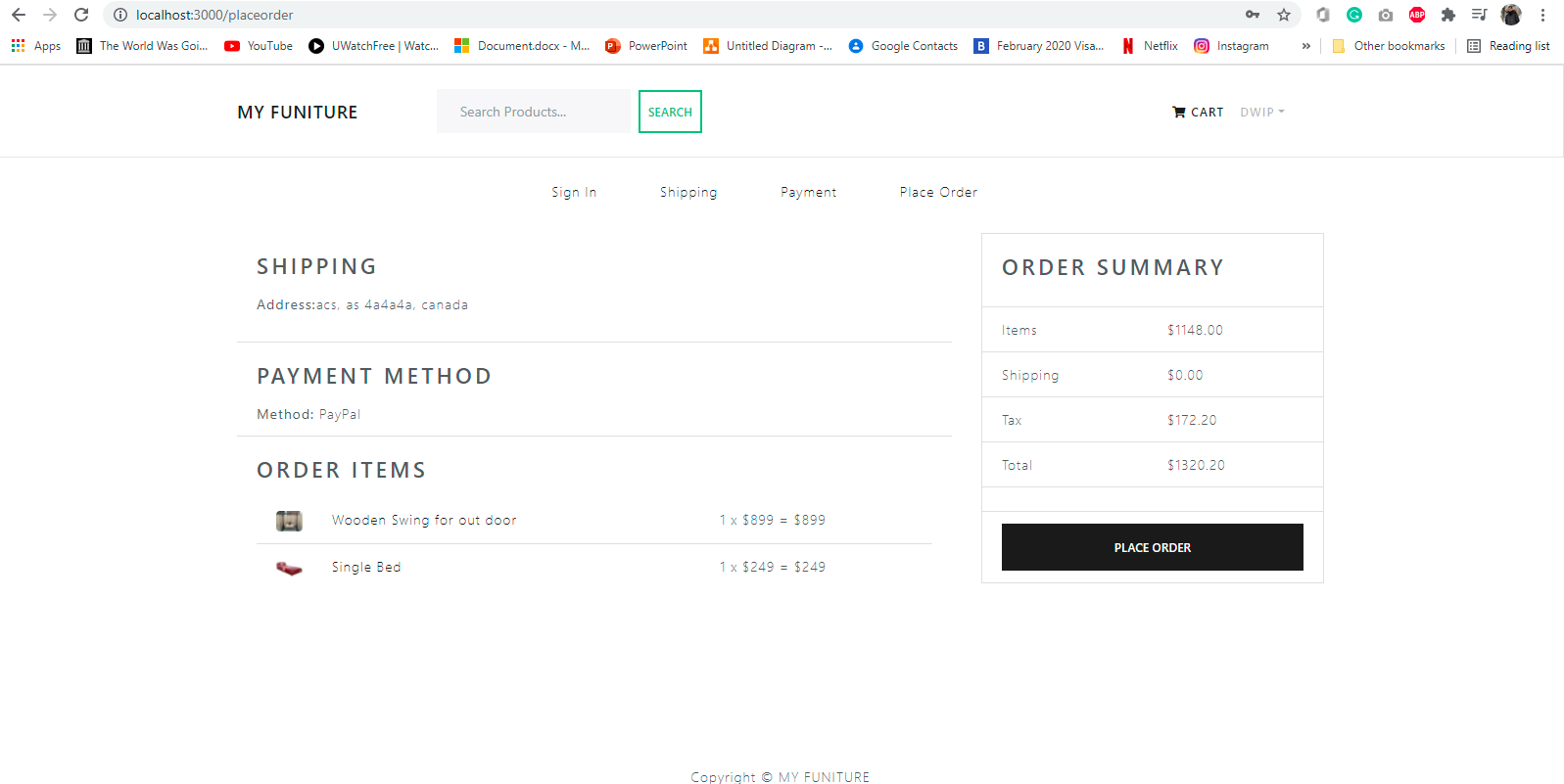
**Cart Page**

****

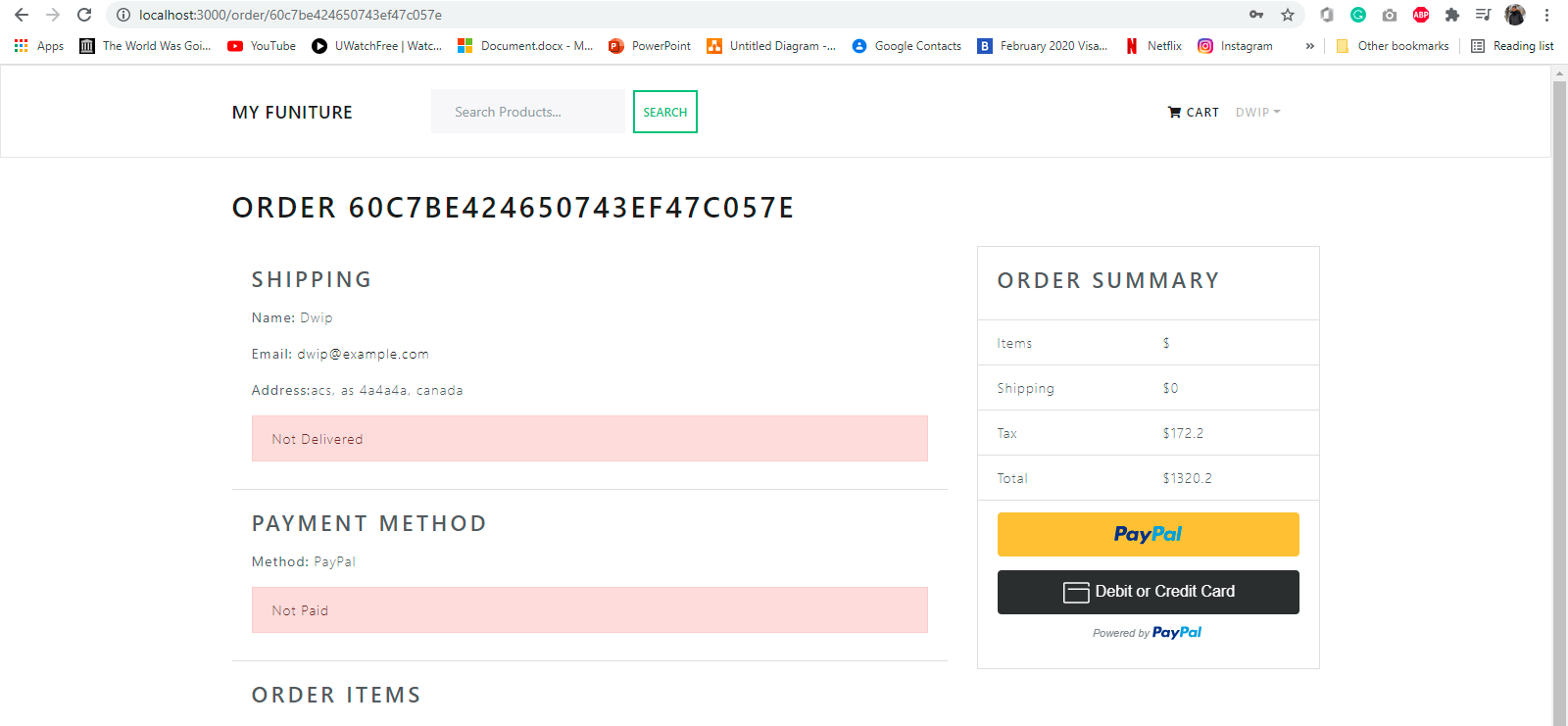
**Shipping details page**

****

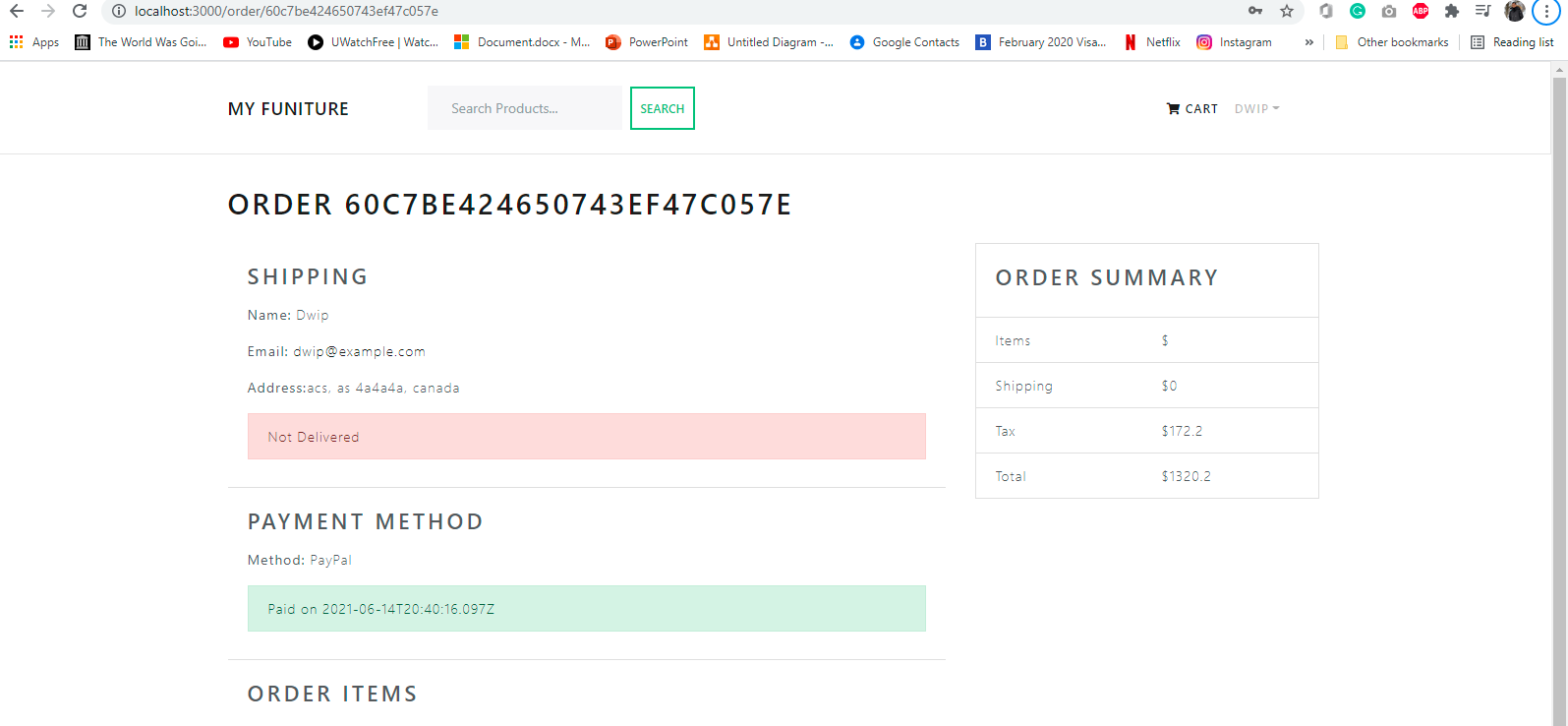
**Conformation page (to conform your product)**

****

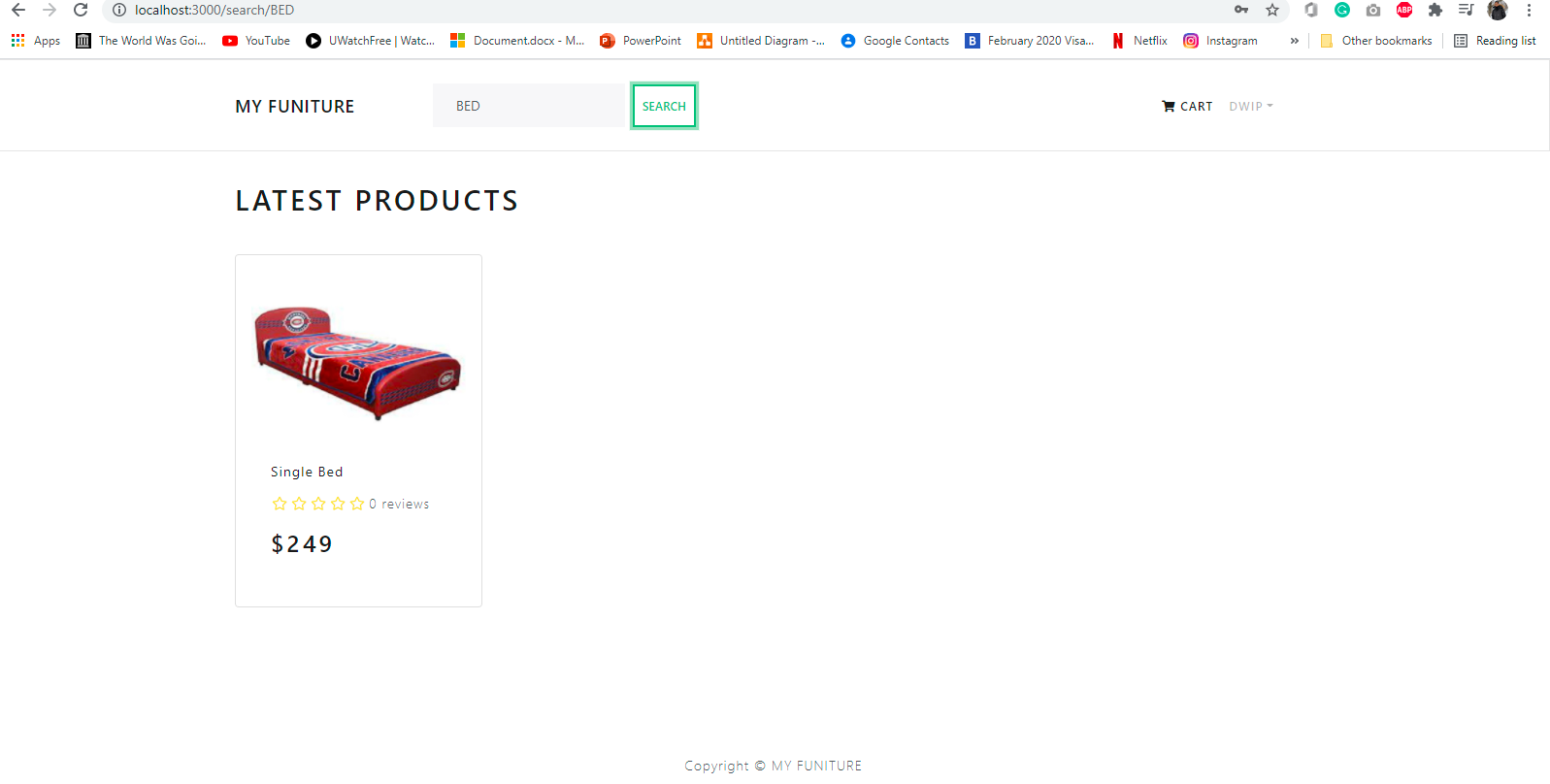
**Payment Page**

****

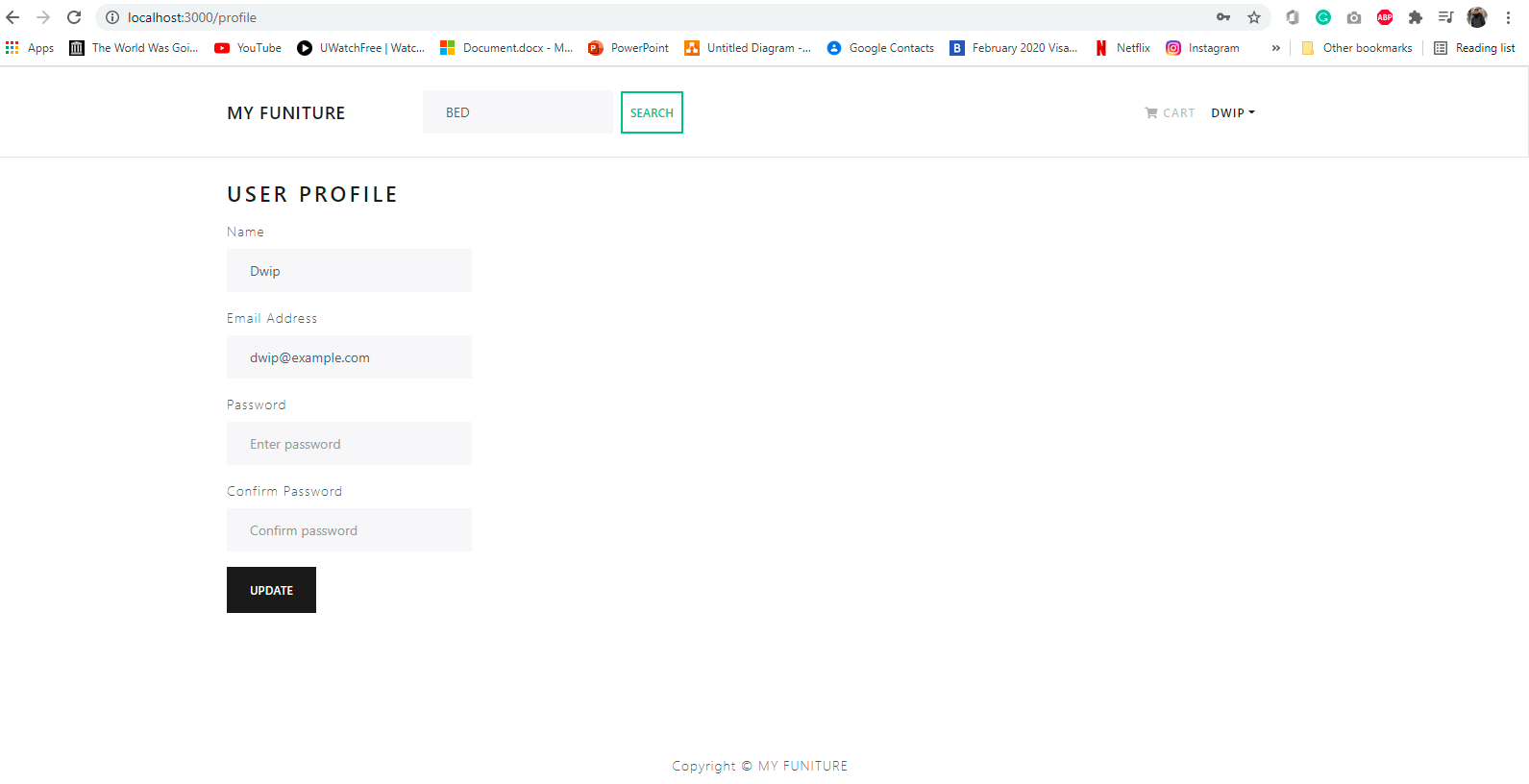
**Payment Page After conformation payment**

****

**Search Bar Function**

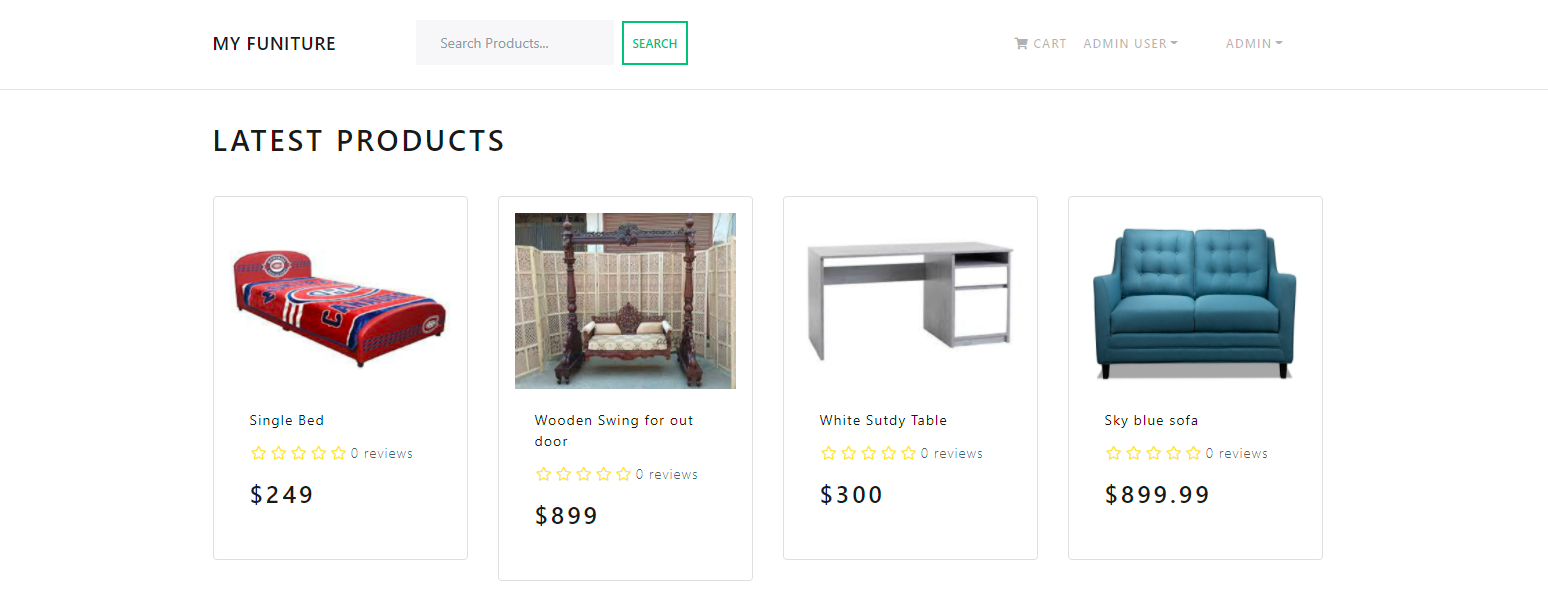
****

**Profile page where we can edit information**

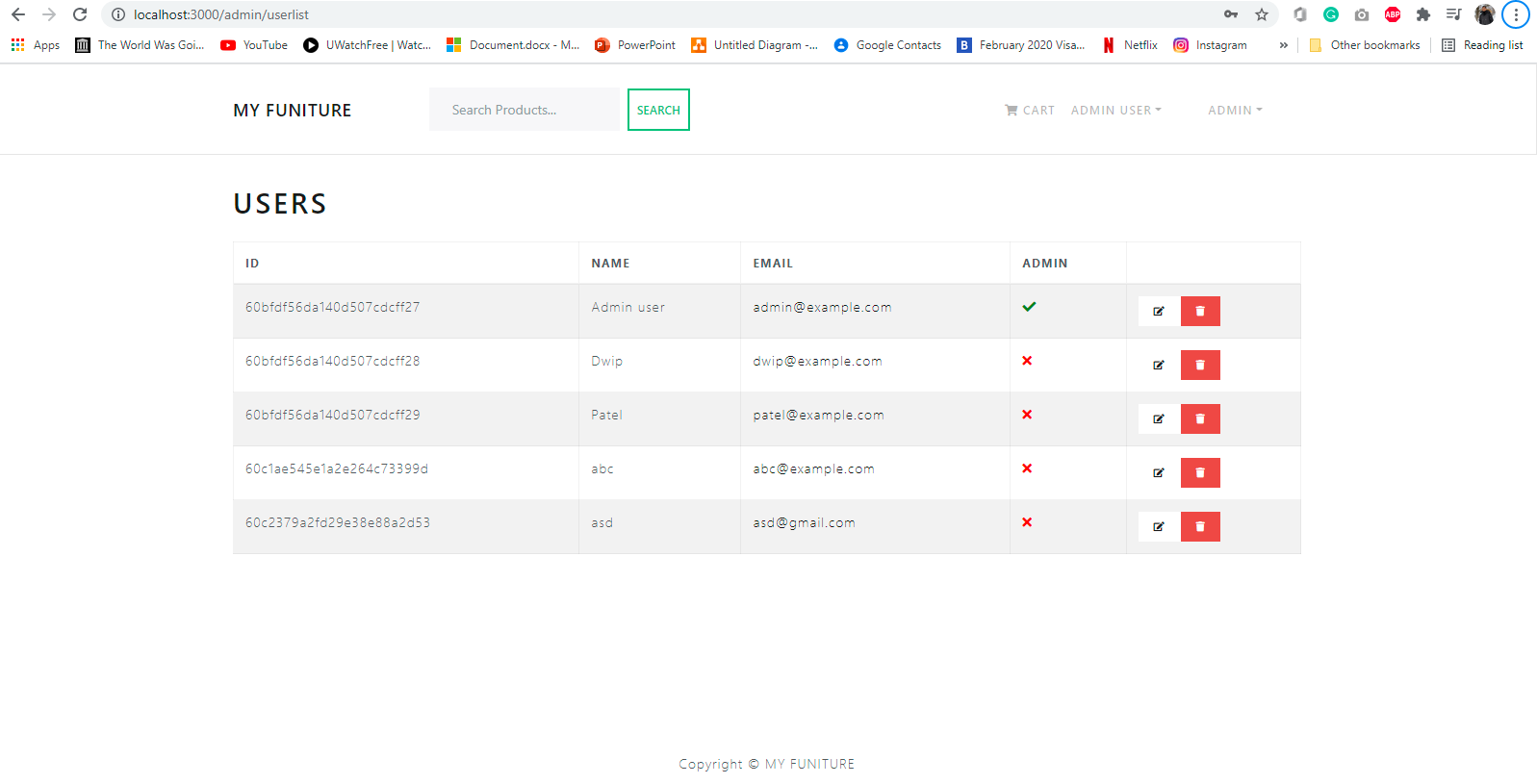
****

# User Interface Admin Side

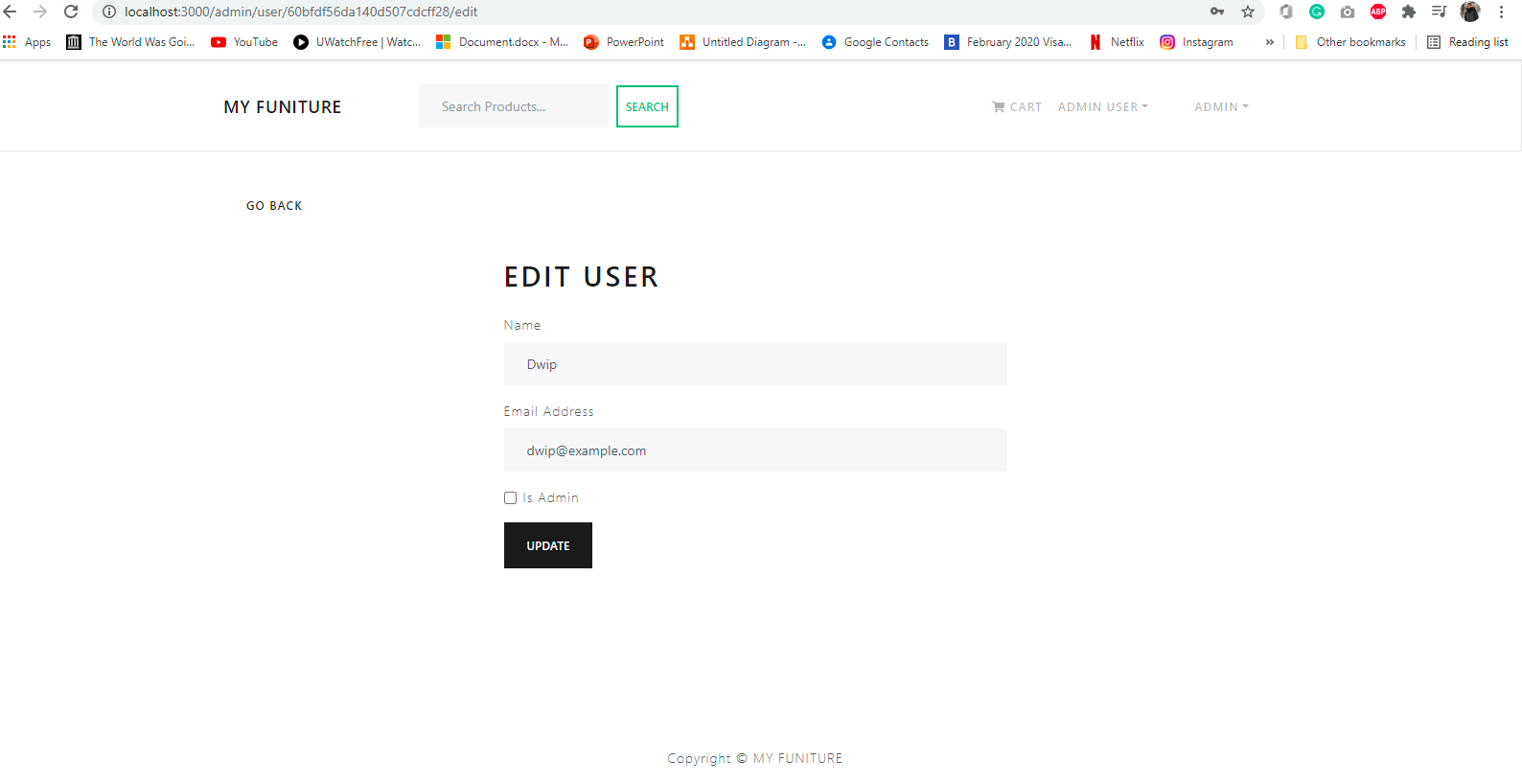
**ADMIN Side Screen short**

****

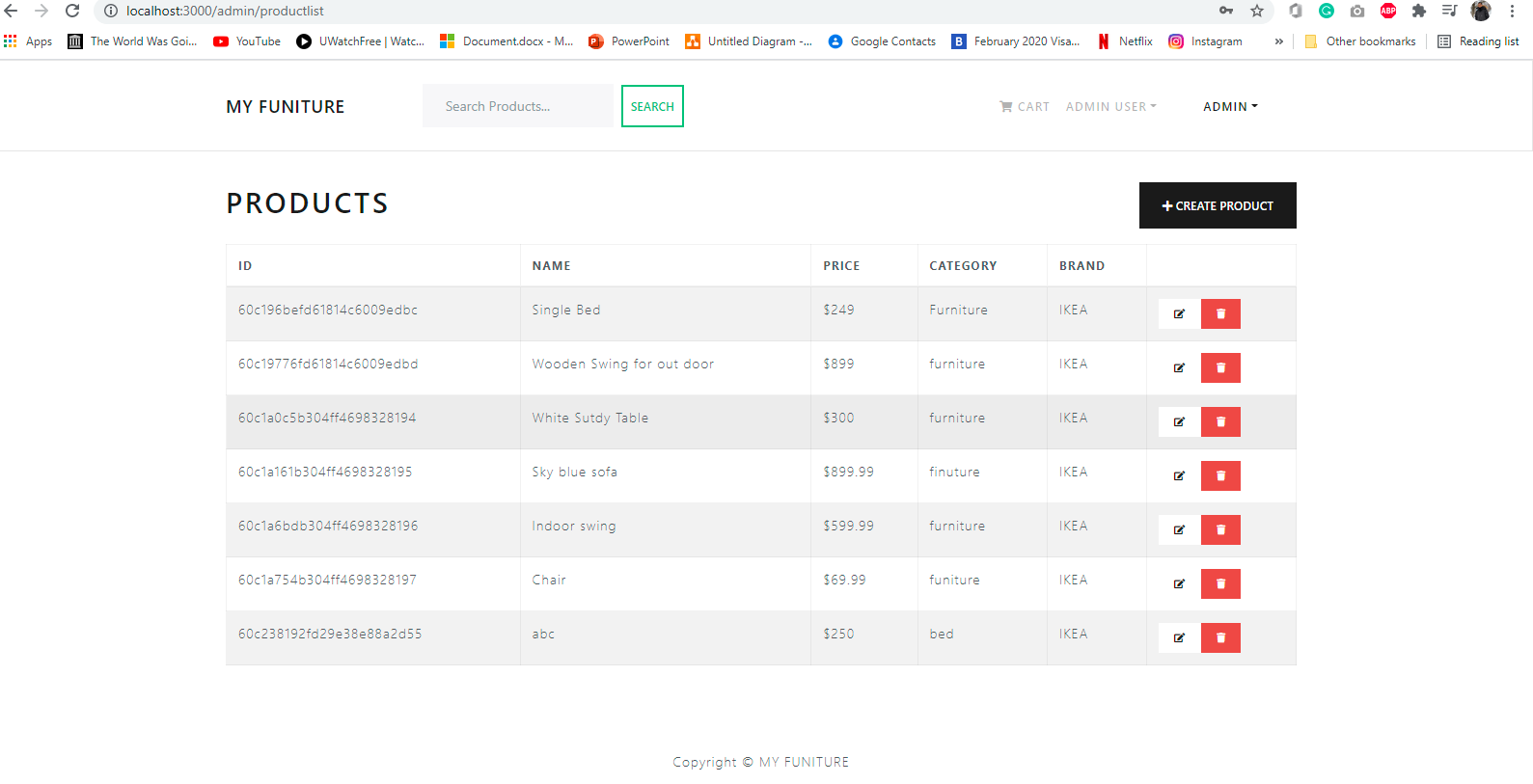
**User section (user details)**

****

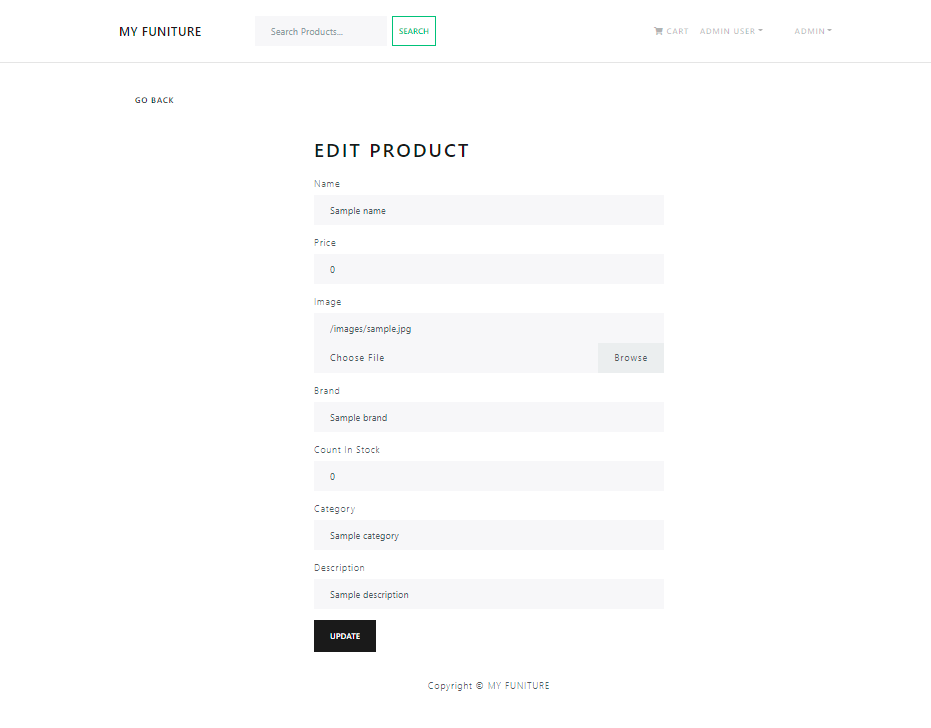
**User Section to update user or make him admin**

****

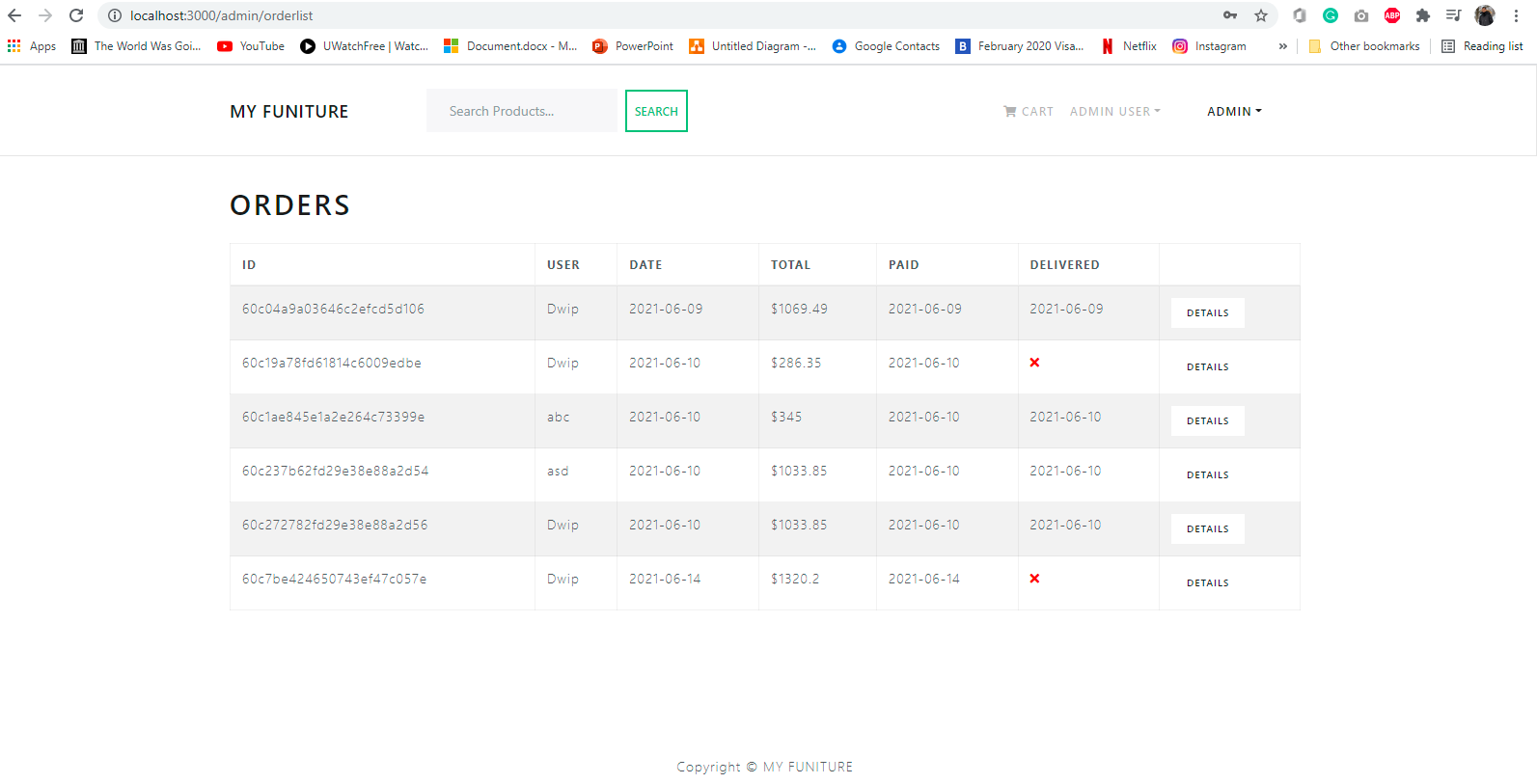
**Product Section with product details**

****

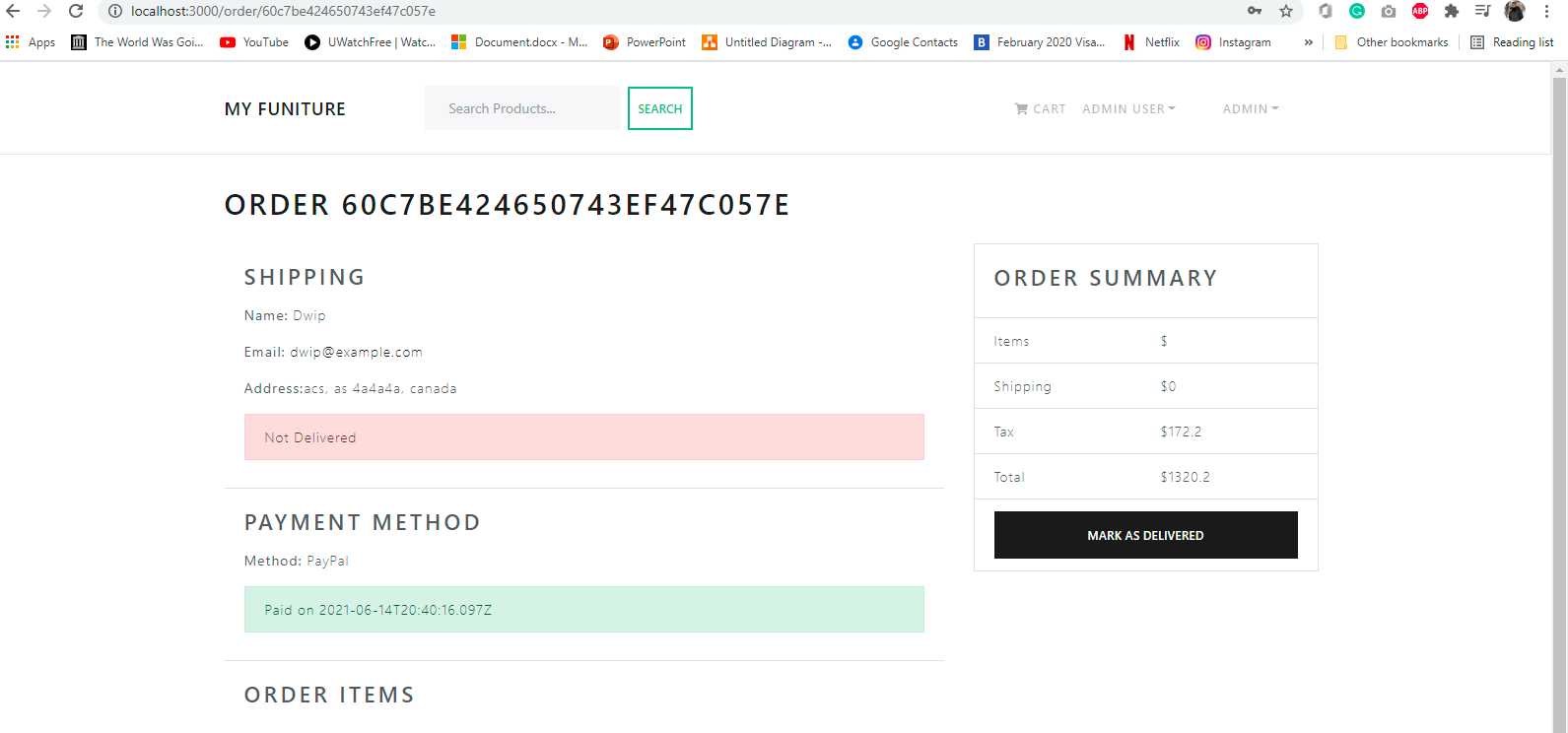
**Add product Page (same page also user for update product)**

****

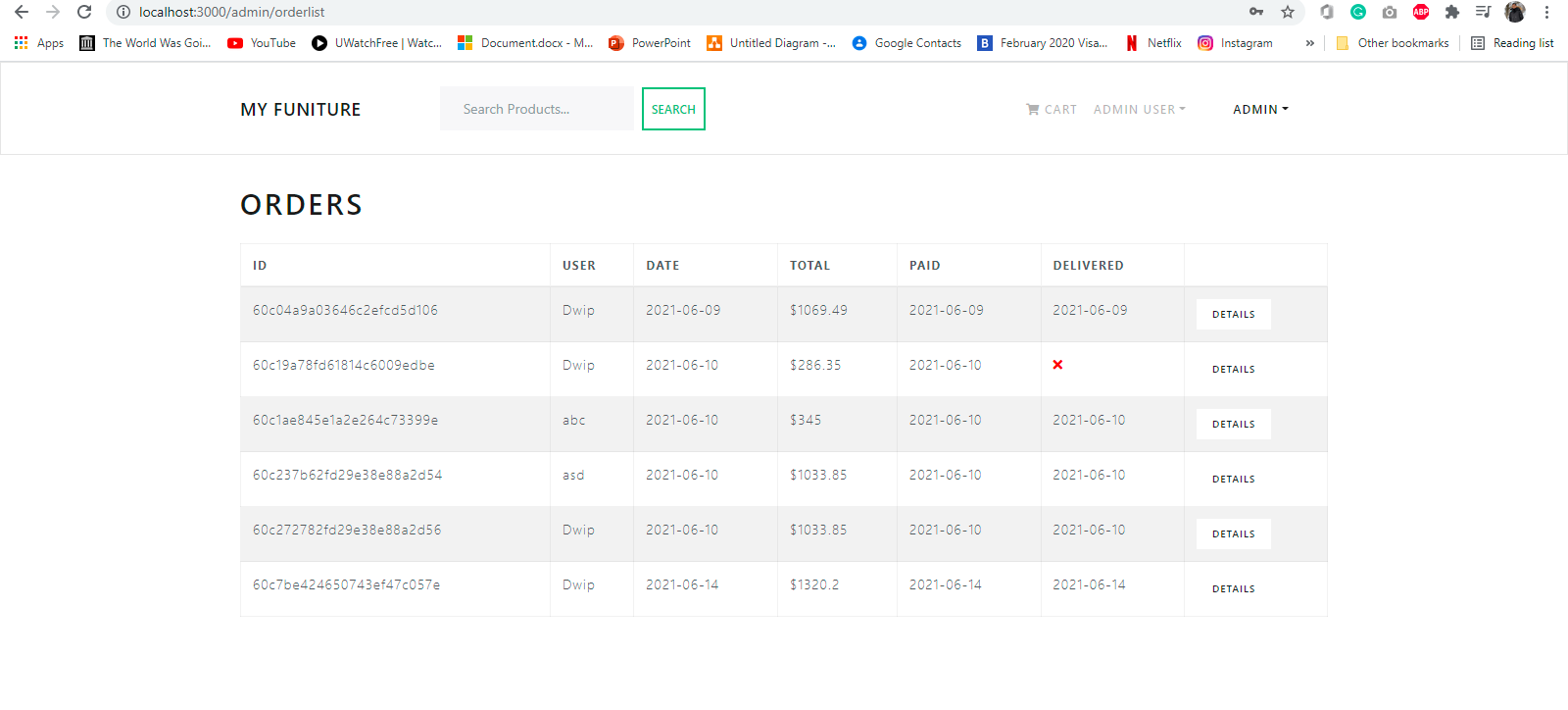
**Order Page (with order details)**

****

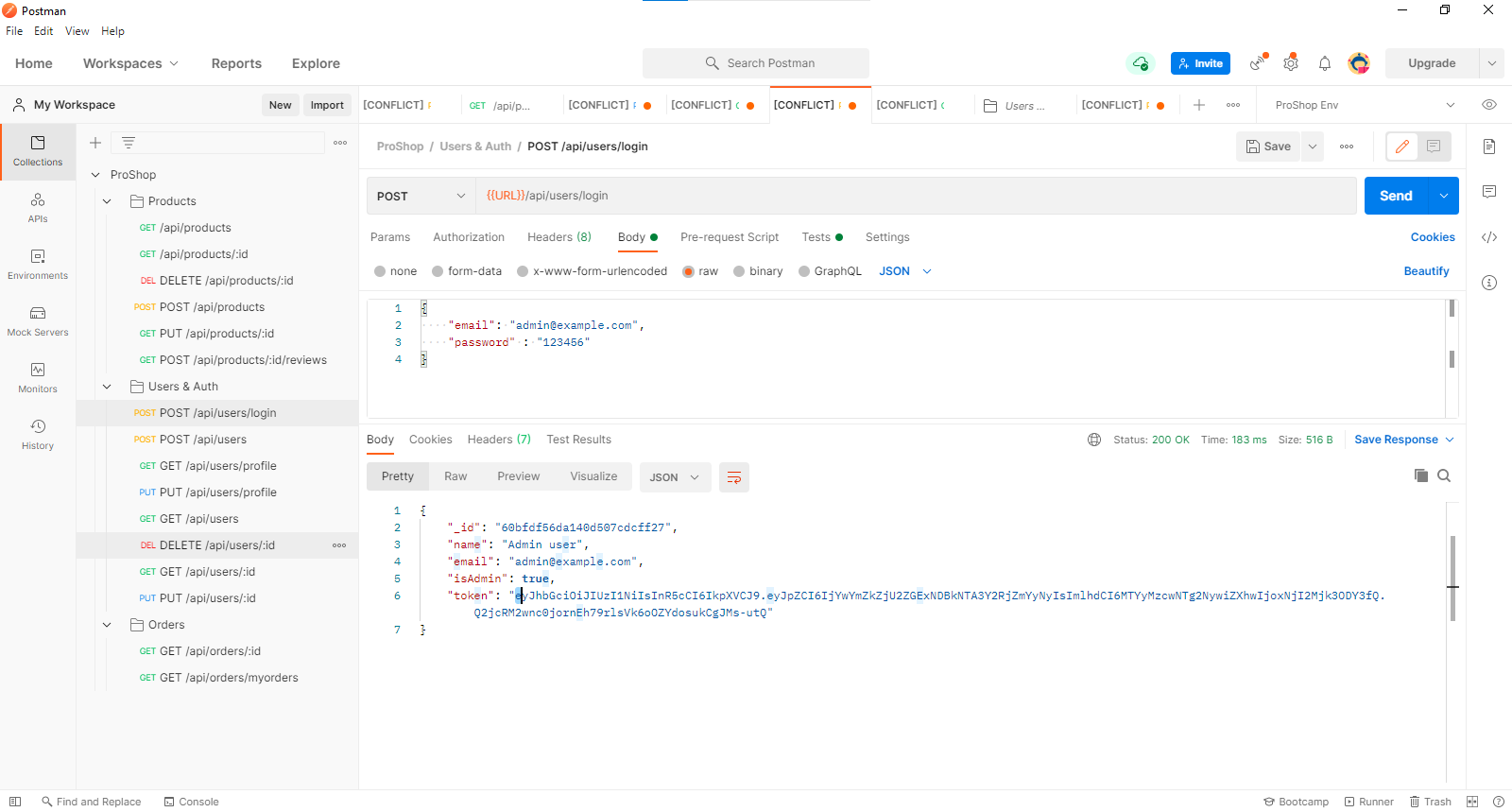
**Order details page(With mark as deliver button)**

****

**Order Details page**

****

# PostMan Screenshot

****

# Conclusion

Overall from this project I have learned so many new things in which I have face many problems but it encourage me to over come an prove me stronger. This ecommerce site almost covers the main work of add, update, delete. Through this project I have learn about post man quite little bit, which may help me in future.

This ecommerce sit is fully working dynamically from backend to frontend.

# References

Axios. (n.d.). From https://www.npmjs.com/package/axios

Body-parser. (n.d.). From https://www.npmjs.com/package/body-parser

Cors. (n.d.). From https://www.npmjs.com/package/cors

Express. (n.d.). From https://www.npmjs.com/package/express

Material-UI/core. (n.d.). From https://www.npmjs.com/package/@material-ui/core

moment. (n.d.). From https://www.npmjs.com/package/moment

Mongoose. (n.d.). From https://www.npmjs.com/package/mongoose

Node. (n.d.). From https://nodejs.dev/learn/introduction-to-nodejs/

Nodemon. (n.d.). *Nodemon*. From https://www.npmjs.com/package/nodemon

React. (n.d.). From https://www.npmjs.com/package/react

react-file-base64. (n.d.). From https://www.npmjs.com/package/react-file-base64

Redux. (n.d.). From https://www.npmjs.com/package/redux

Redux-thunk. (n.d.). From https://www.npmjs.com/package/redux-thunk

Stack\_Overflaw. (n.d.). From https://stackoverflow.com/questions/tagged/node.js?sort=MostVotes&edited=true