Online Marketplace Using MERN STACKS

Project Engineering

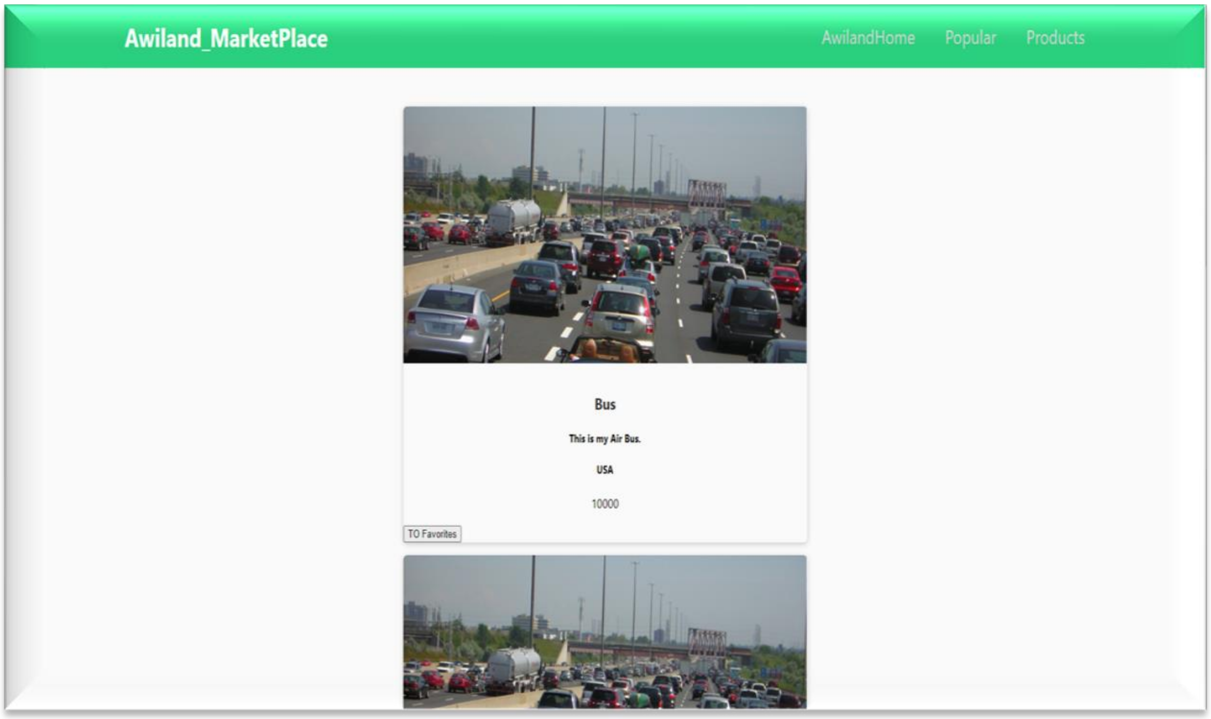
Year 4

Ibrahim Lawal

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Atlantic Technological University Galway

2021/2022

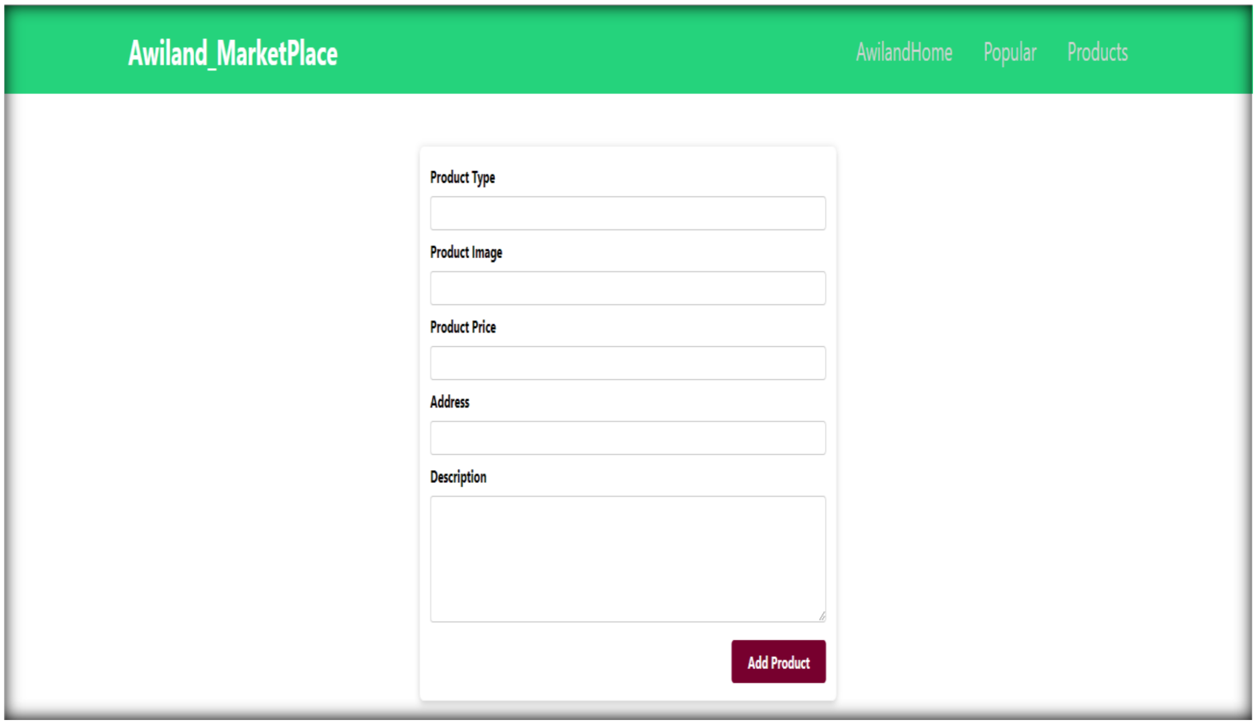


Bus

This is my Air Bus.

USA

10000



Product Type

Product Image

Product Price

Address

Description

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at the Atlantic Technological University Galway.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

__Ibrahim Lawal 12/04/2022__

Acknowledgements

To God be the Glory for the courage, patience, and strength to be able to complete this project. I dedicate this project to the memory my loving brother Amos Anthony Lawal, you will forever be in our hearts.

I acknowledge all my Lecturers, HOD, and staffs for all the support from the start of this project through to fruition, especially Mr Brian O'Shea my project Supervisor, thanks for all the good advice.

A big thank you to my wife Annie Lawal for all the support and encouragement, thank you all and God bless you.

Table of Contents

1 Summary

Awiland Marketplace is an Online Store, I developed because of the inspiration I got from the Nigerian online community during the ban on twitter by the Federal Government of Nigeria, due to the #ENDSARS protests and riots in parts of Nigeria 2020.

The ban meant that small entrepreneurs who mostly use twitter to advertise goods and services couldn't anymore, and therefore were forced to look for VPN or lose business. This was my motivation and inspiration to create an online store where users can post goods and services to gain a prospective customer.

Awiland Marketplace is a MERN STACK online store, created using the MERN STACK technology, which are:

1. **MongoDb**: This is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. [1]

2. **Express**: This is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, you need not worry about low level protocols, processes, etc. [2]

3. **React & NextJs:** Next. js is a widely used framework for building React applications that offer server-side rendering, automatic code-splitting, static exporting options, and easy production builds. It also relieves a lot of the general headaches involved with creating production ready React applications. [3]

4. **NodeJs:** Node. js (Node) is an open-source development platform for executing JavaScript code server-side. [4]

I would be using Node, Express and MongoDB to design the RESTfull APIs and then we would use those APIs in our React frontend. So, basically, it would be a simple Online Marketplace. It is not going to be an elaborate website, but I hope to demonstrate the aim and purpose of the task, while demonstrating the learning and understanding how everything works. I can surely add features on top of this project to make it better. I have kept the design simple and minimal on Frontend side.

I used React Bootstrap to design my React Frontend in a small scale. I aim to be able to connect frontend and backend together for the purpose of sending data and fetching data. So, the features I have in the application that I am building are:

- Authentication using JSON Web Tokens (JWT).
- Option to add, edit, view, and delete all the items in the store.

So, these are the basic features we would be having in our application. Now, let's get familiar with the tech stack we are going to use for building this application.

Frontend — In the frontend side, I am using React as the frontend library, Redux for state management, and React Bootstrap library for basic designing of the interface.

Backend — For the backend side, I have used the Express library on top of Nodejs. MongoDB as the NoSQL database to store data as documents in JSON format, mongoose to connect to my MongoDB database.

I have also created REST APIs with Express and use these endpoints in the React frontend to interact with the backend.

2 Poster

Online MarketPlace Using MERN STACK

Ibrahim Lawal
BEng Software & Electronics Engineering

SUMMARY

Awiland Marketplace is an Online Store, I developed because of the inspiration I got from the Nigerian online community during the ban on twitter by the Federal Government of Nigeria, due to the #ENDSARS protests and riots in parts of Nigeria in the year 2020. The ban meant that small entrepreneurs who mostly use twitter to advertise goods and services couldn't anymore, and therefore were forced to look for VPN or lose their business. This was my motivation and inspiration to create an online store where users can post goods and services to gain a prospective customer. I would be using Node, Express and MongoDB to design the RESTfull APIs and then we would use those APIs in our React frontend. So, basically, it would be a simple Online Marketplace. It is not going to be an elaborate website, but I hope to demonstrate the aim and purpose of the task, while demonstrating the learning and understanding how everything works. I can surely add features on top of this project to make it better. I have kept the design simple and minimal on Frontend side. I would not be dealing with CSS much as our focus would be on understanding how I deal with APIs on the frontend and will focus on the basic aspects. I used React Bootstrap to design my React Frontend minimally. I aim to make a working e-commerce website where everything functions correctly.

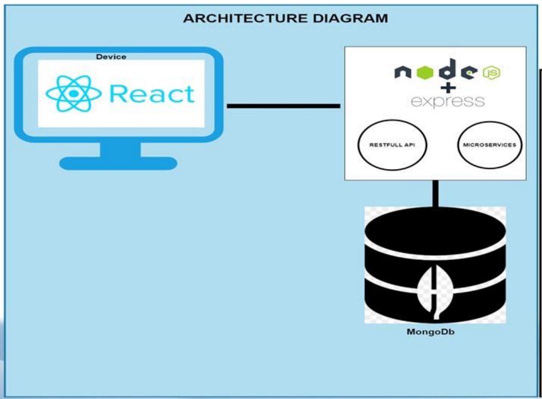
TECHNOLOGIES

mongo DB

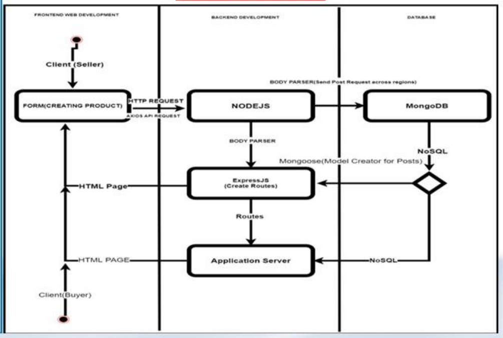
node + express

React JS

ARCHITECTURE DIAGRAM



ACTIVITY DIAGRAM



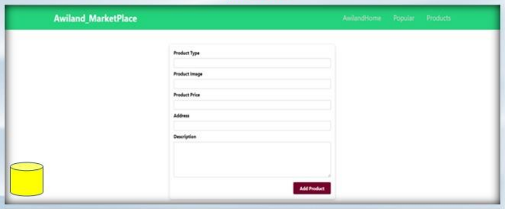
TECHNOLOGIES DESCRIPTION

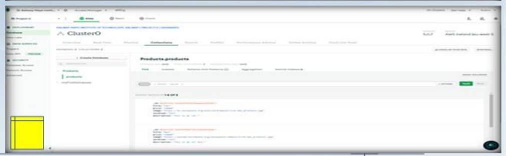
MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB
Node. js (Node) is an open source development platform for executing JavaScript code server-side.
ReactJS is JavaScript library used for building reusable UI components
.

RESULTS DESCRIPTION

- Product Page: Allows users to create a product that will be displayed on the Awiland HomePage.
- MongoDb CloudAtlas Database showing collections of product created with each product having a unique key.

RESULTS





3 Introduction

AwilandMarketplace.com is an online marketplace that I created using MERN STACK, to allow users who want to advertise their products to post images, address, product description and price, to their customers.

The inspiration behind this project is the Twitter ban by the Nigerian Government, after the youth protested police brutality and bad governance in Nigeria. The ban meant that small scall entrepreneurs that use Twitter as their marketing platform have no other platform to reach their customers.

AwilandMarketplace.com will allow users to post products for customers to view and contact the marketer.

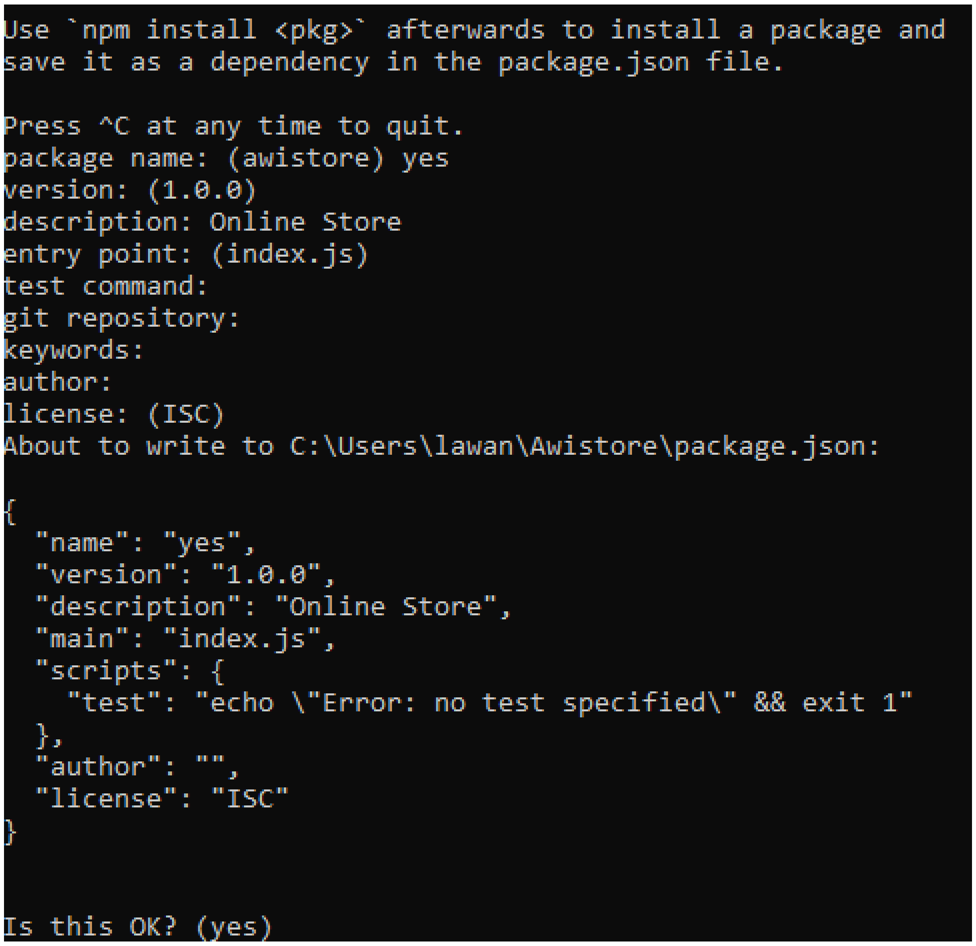
The scope of this project is MERN Stack is one of the simplest frameworks for building different types of applications and websites. It is one of the most effective and reliable internet stacks in 2021. MERN is a free open-source JavaScript software stack for building dynamic internet websites and internet applications. The stack is all about JavaScript from the front and to the rear end.

4 Background

To get started with this project, I downloaded and installed Nodejs into my system, this will allow me to use NPM (Node Package Manager). Then I opened a terminal and cd into the folder where I want to have the project created, created a new folder, and named it awistore. Then move into the created folder and type in the following command in the terminal to start a new Node project there.

```
npm init
```

A series of question will be displayed on the terminal like this:



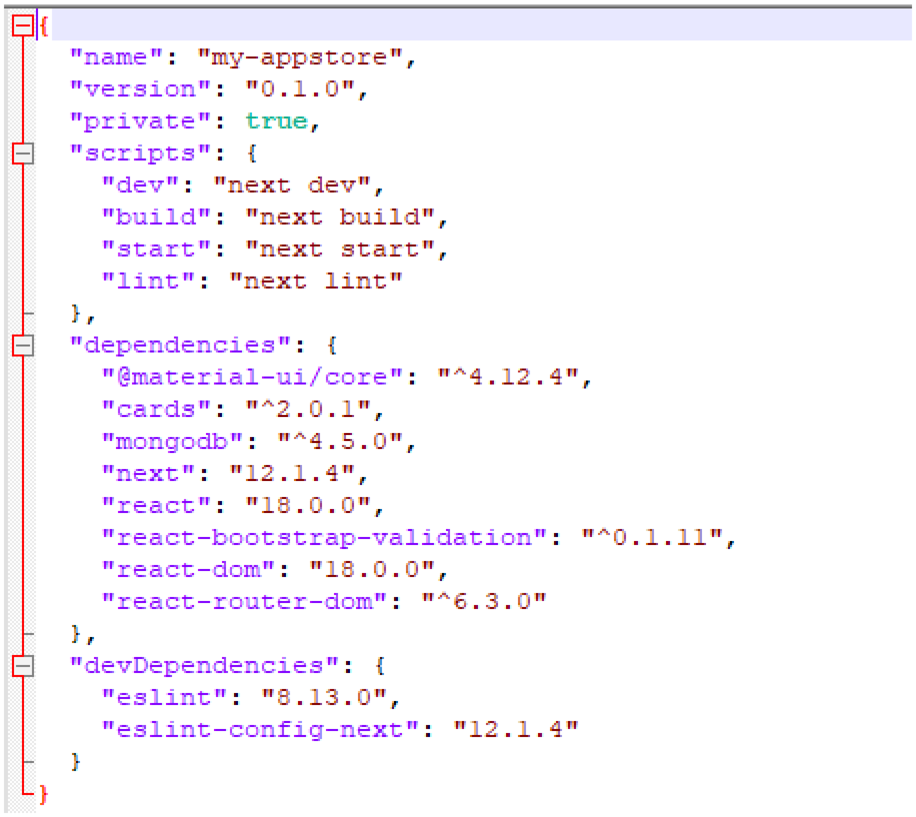
```
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (awistore) yes
version: (1.0.0)
description: Online Store
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\lawan\Awistore\package.json:

{
  "name": "yes",
  "version": "1.0.0",
  "description": "Online Store",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)
```

I choose the name for my package and gave it a description, then put my name in the author section. change the entry point from index.js to server.js as I am going to name my entry file as server.js instead of index.js. It will work like a server so naming it as such seems more reasonable. I left all the other fields blank.

When I click on yes, it then would create a package.json file in that folder. I then opened the package.json file in my code editor, which is VS Code (Visual Studio Code).

I would now need to install certain dependencies using npm which would then automatically add them as dependencies in my package.json file.

Here is the package.json file with all the dependencies I am using for this project as of now. I will add some dependencies later when I need to do so.



```
{
  "name": "my-appstore",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "@material-ui/core": "^4.12.4",
    "cards": "^2.0.1",
    "mongodb": "^4.5.0",
    "next": "12.1.4",
    "react": "18.0.0",
    "react-bootstrap-validation": "^0.1.11",
    "react-dom": "18.0.0",
    "react-router-dom": "^6.3.0"
  },
  "devDependencies": {
    "eslint": "8.13.0",
    "eslint-config-next": "12.1.4"
  }
}
```

Dependencies:

- @material-ui/core: This is simply a library that allows us to import and use different components to create a user interface in our React applications. [5]
- Cards: Cards contain content and actions about a single subject. Material UI for React has this component available for us, and it is very easy to integrate. We can use the Card Component in ReactJS using the following approach.

  Creating React Application and Installing Module:
  Step 1: Create a React application using the following command.
  npx create-react-app folder name

  Step 2: After creating your project folder i.e. folder name, move to it using the following command.
  cd folder name

  Step 3: After creating the ReactJS application, Install the material-ui modules using the following command.
  npm install @material-ui/core [6]

- MongoDb: This is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. [1]
- NextJs: is a JavaScript framework created by Zeit. It lets you build server-side rendering and static web applications using React. It's a great tool to build your next website. It has many great features and advantages, which can make NextJs your first option for building your next web application. [7]
- ReactJs: React components implement a render() method that takes input data and returns what to display. [8]
- React Bootstrap Validation Form: This allows us to update the form fields to useState, check and validate the data that has been entered, if there are problems with the data stop the submission and display error messages, and if the data is ok, handle the onClick event of the submit button and submit the form.
- ReactDOM: The react-dom package provides DOM-specific methods that can be used at the top level of your app and as an escape hatch to get outside the React model if you need to:- import * as ReactDOM from 'react-dom'; [9]
- React Router DOM: React Router DOM is a npm package that enables you to implement dynamic routing in a web app. It allows you to display pages and allow users to navigate them. [10]

I have added a few scripts to make it easy to run the server and client. They are:

start — It uses node to run the server.js file. It would need to restart for updates.
server — It uses nodemon to run the server.js file which allows it to update changes and restart the server automatically.
client — Running this command runs the client. I used a prefix to let it know that we want to first move into client folder and then run the command.
dev — It is used concurrently to run both the server and client at the same time.

Next, I created a **server.js** file in the root directory, I started by doing all the required imports of the various libraries I will need in this file.

```
const express = require('express');
const mongoose = require('mongoose');
const path = require('path');
const config = require('config');
```

I then call our express app and will set it to use it in our application.

```
const app = express();
app.use(express.json());
```

I will set up my server file to serve static content which will be generated from React app in production. This will only work in the production environment.

```
if(process.env.NODE_ENV === 'production') {
  app.use(express.static('client/build'));
  app.get('*', (req, res) => {
res.sendFile(path.resolve(__dirname,'client','build','index.html'));
  });
}
```

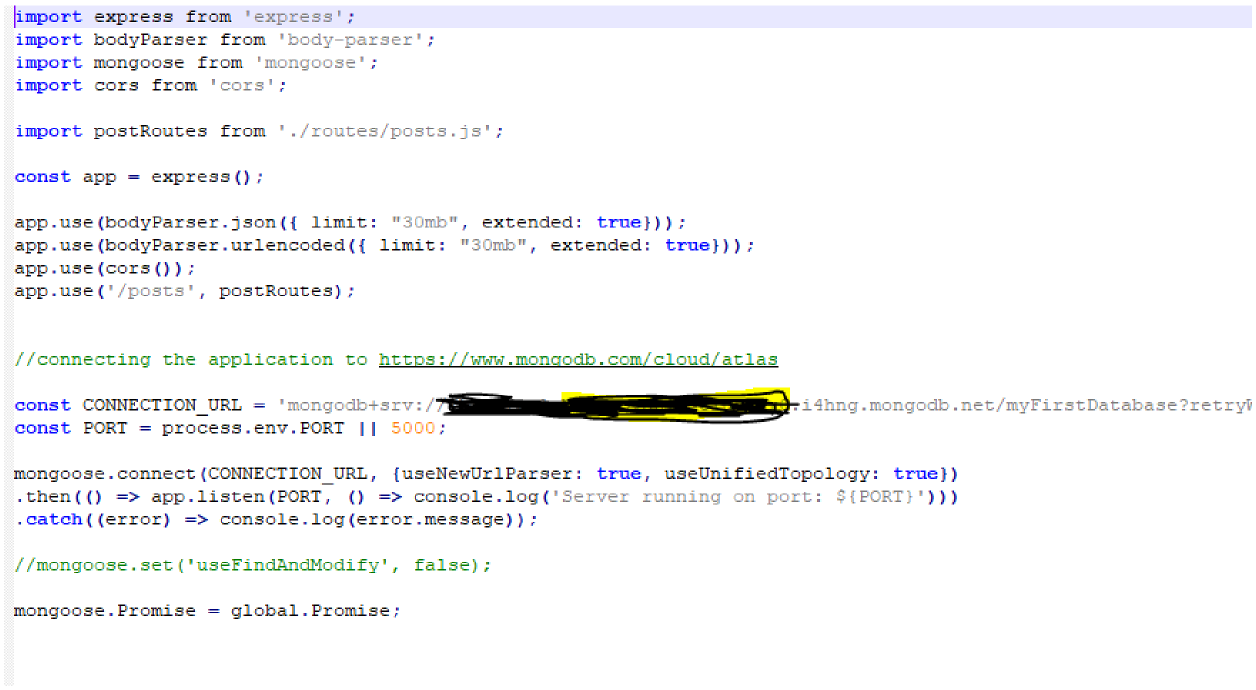
I configured my server file to connect to the MongoDB database and then start running the server to listen to our requests on port 8000.

```
const dbURI = config.get('dbURI');
const port = process.env.PORT || 8000;
mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true,
useCreateIndex:true })
  .then((result) => app.listen(port))
  .catch((err) => console.log(err));
```

I have used *config* to get my Database URI. I define a port variable to use any port value present in the environment variable as in case of production but in development, I will use port 8000.

I then connect to my database using *mongoose* and after I have successfully connected to the database, I start listening to requests on the port i.e., server is up and running.

This is the server.js file which I have built till now: -



```
import express from 'express';
import bodyParser from 'body-parser';
import mongoose from 'mongoose';
import cors from 'cors';

import postRoutes from './routes/posts.js';

const app = express();

app.use(bodyParser.json({ limit: "30mb", extended: true}));
app.use(bodyParser.urlencoded({ limit: "30mb", extended: true}));
app.use(cors());
app.use('/posts', postRoutes);


//connecting the application to https://www.mongodb.com/cloud/atlas

const CONNECTION_URL = 'mongodb+srv://[redacted]i4hng.mongodb.net/myFirstDatabase?retryW
const PORT = process.env.PORT || 5000;

mongoose.connect(CONNECTION_URL, {useNewUrlParser: true, useUnifiedTopology: true})
.then(() => app.listen(PORT, () => console.log('Server running on port: ${PORT}')))
.catch((error) => console.log(error.message));

//mongoose.set('useFindAndModify', false);

mongoose.Promise = global.Promise;
```

5 Project Architecture

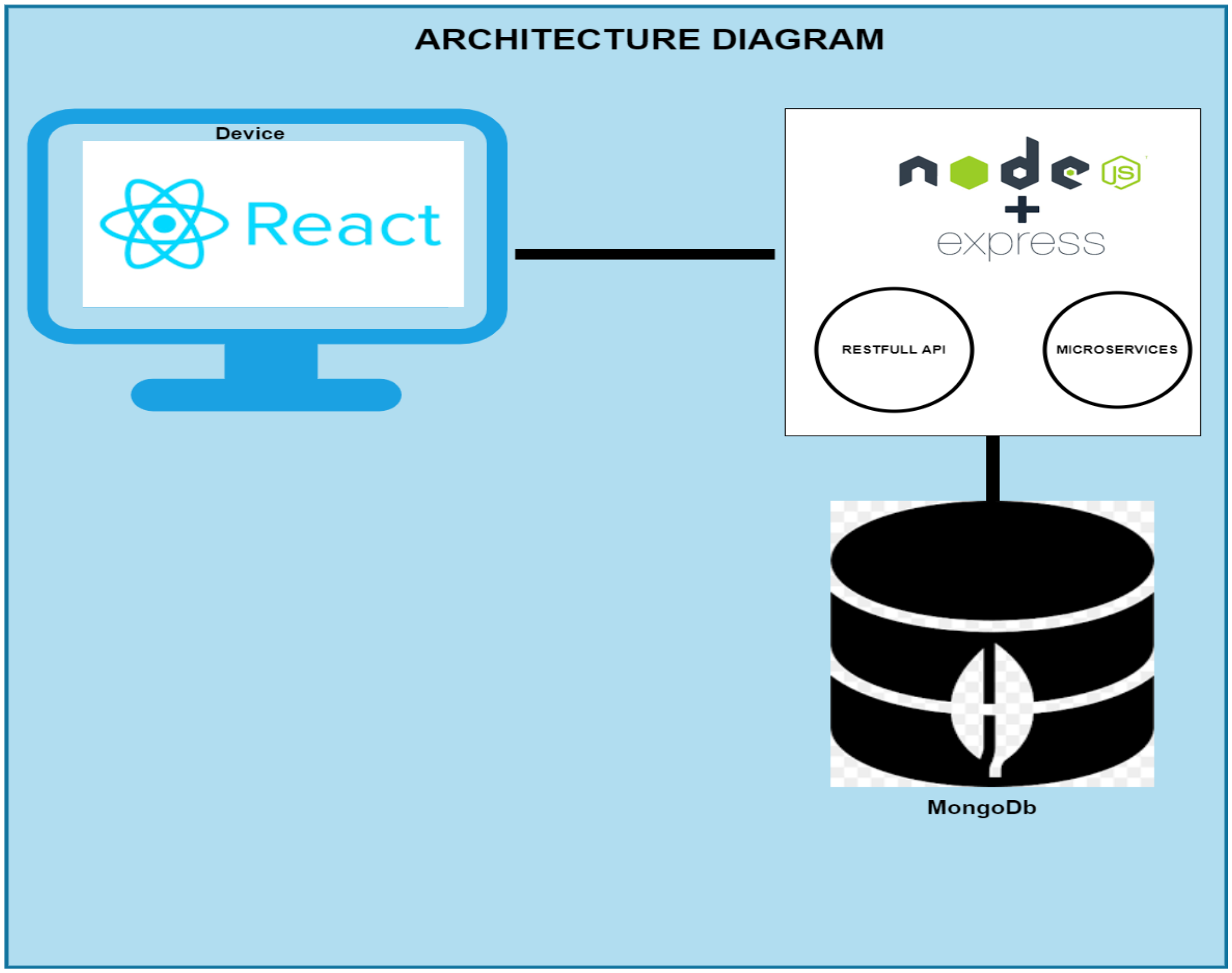
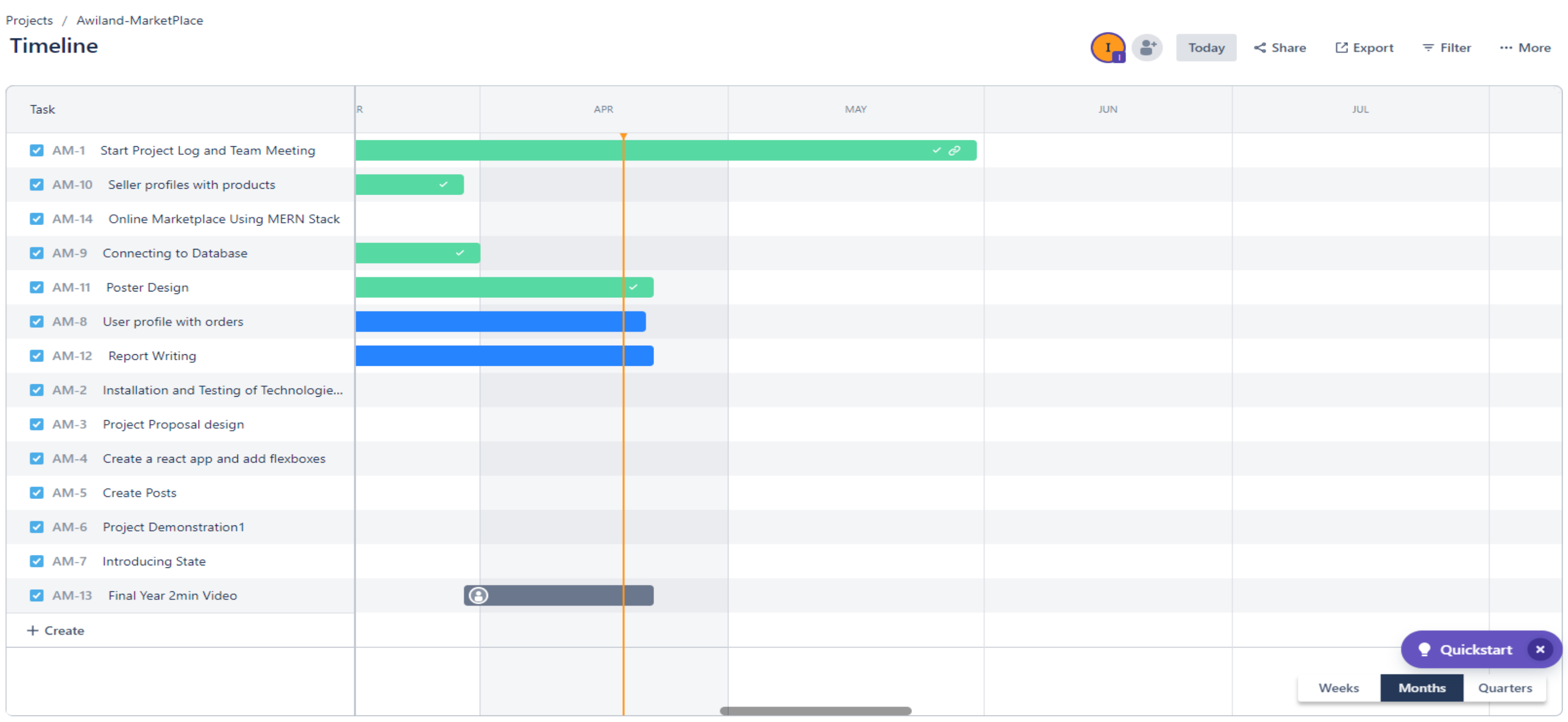


Figure 1 Architecture Diagram

MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB
Nodejs (Node) is an open-source development platform for executing JavaScript code server-side.
ReactJS is JavaScript library used for building reusable UI components
RestFul API is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services
Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs

6 Project Plan

For my project planning and management, I used a project managing software called JIRA, and my project logs to help me to keep track of my tasks and set reminders.



Projects / Awiland-MarketPlace

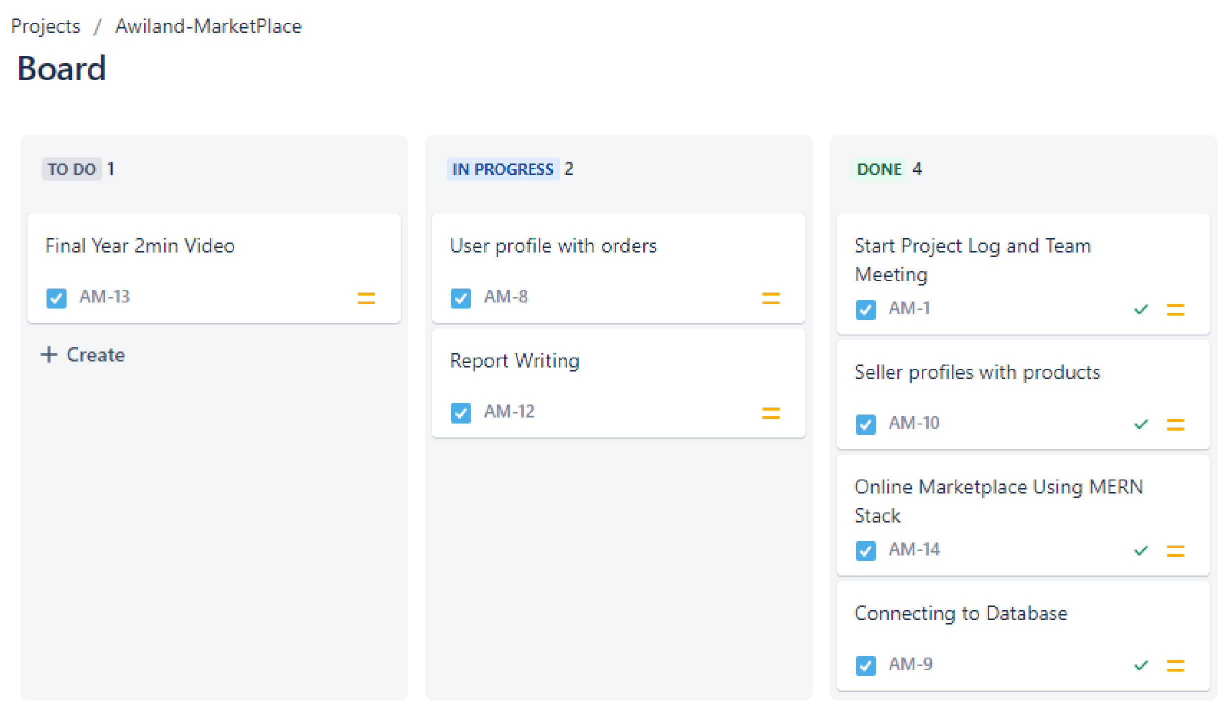
Timeline

Today Share Export Filter More

Task	
AM-1	Start Project Log and Team Meeting
AM-10	Seller profiles with products
AM-14	Online Marketplace Using MERN Stack
AM-9	Connecting to Database
AM-11	Poster Design
AM-8	User profile with orders
AM-12	Report Writing
AM-2	Installation and Testing of Technologie...
AM-3	Project Proposal design
AM-4	Create a react app and add flexboxes
AM-5	Create Posts
AM-6	Project Demonstration1
AM-7	Introducing State
AM-13	Final Year 2min Video

+ Create

Figure 2 Project Timeline displaying progress of each task.


Projects / Awiland-MarketPlace

Board

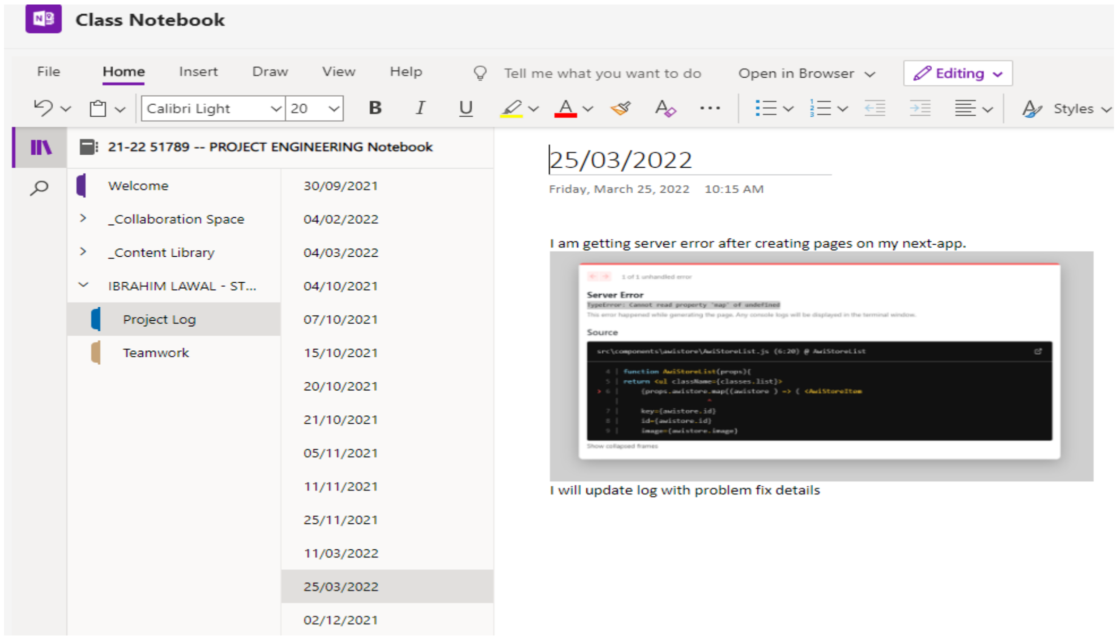
TO DO 1
- Final Year 2min Video — AM-13
- + Create

IN PROGRESS 2
- User profile with orders — AM-8
- Report Writing — AM-12

DONE 4
- Start Project Log and Team Meeting — AM-1
- Seller profiles with products — AM-10
- Online Marketplace Using MERN Stack — AM-14
- Connecting to Database — AM-9

Figure 3 Project Management Board, displaying to do lists, in progress list, and finished tasks list.

ClassNotebook:



21-22 51789 -- PROJECT ENGINEERING Notebook

Welcome	30/09/2021
_Collaboration Space	04/02/2022
_Content Library	04/03/2022
IBRAHIM LAWAL - ST...	04/10/2021
Project Log	07/10/2021
Teamwork	15/10/2021
	20/10/2021
	21/10/2021
	05/11/2021
	11/11/2021
	25/11/2021
	11/03/2022
	25/03/2022
	02/12/2021

25/03/2022

Friday, March 25, 2022 10:15 AM

I am getting server error after creating pages on my next-app.

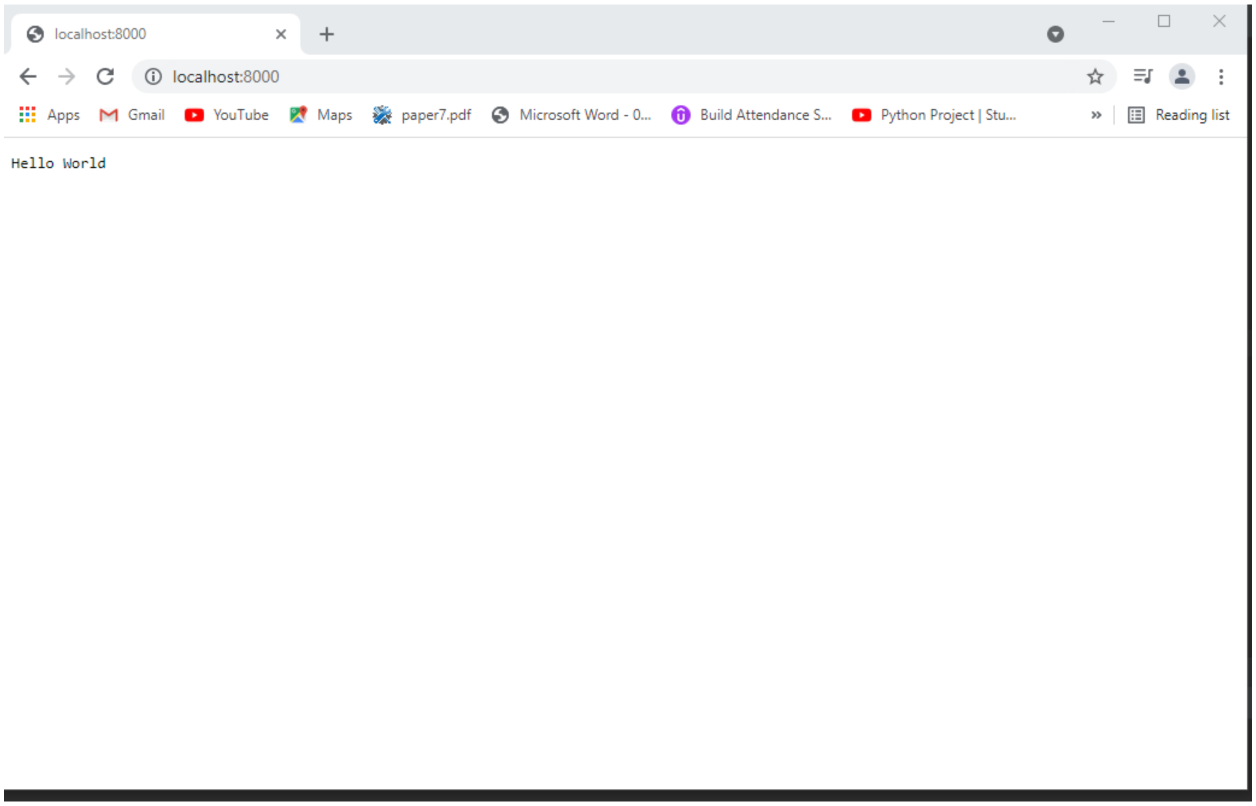
```
1 of 1 unhandled error

Server Error
TypeError: Cannot read property 'map' of undefined
This error happened while generating the page. Any console logs will be displayed in the terminal window.

Source
src\components\awistore\AwiStoreList.js (6:20) @ AwiStoreList

function AwiStoreList(props){
return <ul className={classes.list}>
    {props.awistore.map((awistore ) => ( <AwiStoreItem
                  ^
    key={awistore.id}
    id={awistore.id}
    image={awistore.image}

Show collapsed frames
```

I will update log with problem fix details

Figure 4 Project logs showing input date and tasks.

7 AwilandMarketPlace.com

This is an online marketplace created using MERN Stack, to begin the development phase, I created my first NodeJs server, my first application, and then ran it and the result below:



```
Hello World
```

Figure 5 My first Application.

Then I started designing my architecture diagram for my project proposal, created the documents needed to complete this task. Attending team meetings to have my work peer reviewed by my team members.

Created my Homepage using ReactJs, first I import React library, import the .css file, and then the product component. Added a function that returns an HTML page, below is my first JavaScript code Home.js:

```
import React from 'react';
import "./Home.css";
//import Product from './Product';
function Home() {
  return (
    <div className='home'>
      <div className="home__container">
      <img
       className="home__image" src=""
       alt=""/>
       <div className="home__row">
       </div>
    </div>
  </div>
  );
}
export default Home
```

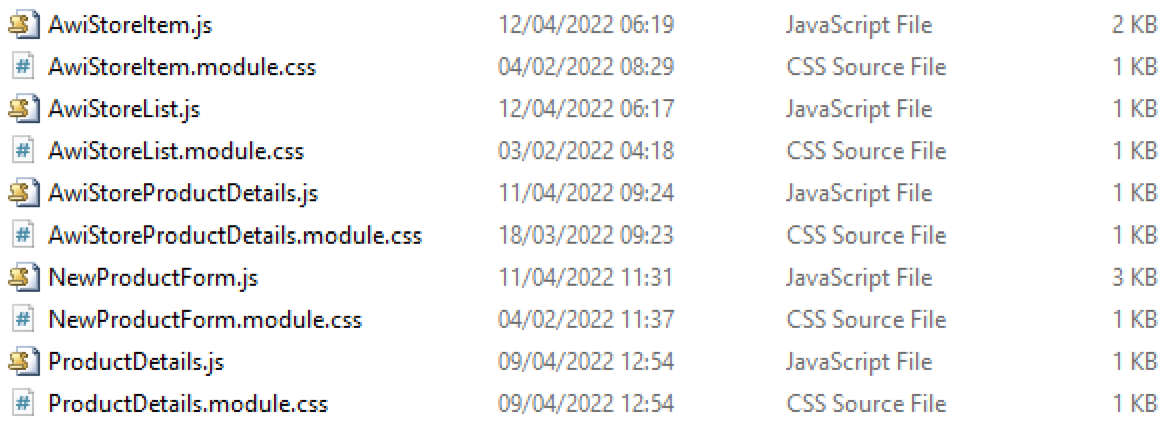
And then a Home.css file to style the HTML page, the css describes how my HTML elements should be displayed. Below is the code:

```
.home__image {
    width: 100%;
    z-index: -1;
    margin-bottom: -150px;
    /*mask-image: linear-gradient(to bottom, rgba(0,
0, 0, 1), rgba(0, 0, 0, 0));*/
}
```

Next I added a Header.js file, where I started by importing React library from React, my header.css file for styling, a SearchIcon component from @mui/icons-material/Search. I also imported a ShoppingCartIcon, AccountCircleIcon, and NotificationNoneSharpIcon, from @mui/icons-materials. Created a function Header() that returns my HTML page, code below:

```
import React from 'react';
import './Header.css';
import SearchIcon from '@mui/icons-material/Search';
import ShoppingCartIcon from '@mui/icons-material/ShoppingCart';
import AccountCircleIcon from '@mui/icons-material/AccountCircle';
import NotificationsNoneSharpIcon from '@mui/icons-material/NotificationsNoneSharp';
function Header() {
  return (
    <div className='header'>
      <img className="header__logo" src="https://paradacreativa.es/wp-
content/uploads/2020/08/Como-activar-Marketplace-en-Facebook.jpg"/>
     <div
      className="header__search">
       <input className="header__searchInput" type="text"/>

      <SearchIcon className="header__searchIcon"/>
      </div>
      <div className="header__nav">
      </div>
      <div className='header__option'><AccountCircleIcon />

        { /*<span className='header__optionLineOne'>Guest</span>

         <span className='header__optionLineTwo'>Marketplace</span>*/}

      </div>

      <div className='header__optionNotification'><NotificationsNoneSharpIcon />

         {/*<span className='header__optionLineOne'>Returns</span>

         <span className='header__optionLineTwo'>& Orders</span>*/}

      </div>

      <div className="header__optionCart"><ShoppingCartIcon />

      <span className="header__optionLineTwo header__cartCount">0</span>

      </div>

      </div>

  );

}

export default Header;
```

Components: At this I have moved all my files from ReactJs to NextJs. I added 3 components folders, awistore, layout, and ui. The awistore component contains JavaScript files and module.css files, used to design how the pages are displayed. Below is a screen shot of the contents of the awistore component.



AwiStoreItem.js	12/04/2022 06:19	JavaScript File	2 KB
AwiStoreItem.module.css	04/02/2022 08:29	CSS Source File	1 KB
AwiStoreList.js	12/04/2022 06:17	JavaScript File	1 KB
AwiStoreList.module.css	03/02/2022 04:18	CSS Source File	1 KB
AwiStoreProductDetails.js	11/04/2022 09:24	JavaScript File	1 KB
AwiStoreProductDetails.module.css	18/03/2022 09:23	CSS Source File	1 KB
NewProductForm.js	11/04/2022 11:31	JavaScript File	3 KB
NewProductForm.module.css	04/02/2022 11:37	CSS Source File	1 KB
ProductDetails.js	09/04/2022 12:54	JavaScript File	1 KB
ProductDetails.module.css	09/04/2022 12:54	CSS Source File	1 KB

The layout component contains JavaScript files and module.css for styling. The layout library allows me to define the interface structure and organise content inside the collapsible or expandable cells. Below is the code:

```
import MainNavigation from './MainNavigation';
import classes from './Layout.module.css';
function Layout(products) {
  return (
    <div>
      <MainNavigation />
      <main
className={classes.main}>{products.children}</mai
n>
    </div>
  );
}
export default Layout;
```

The MainNavigation as the name suggest containing JavaScript file that holds HTML links used for navigating between pages. Below is the code.

```
//import {Link} from 'next/link';
import Link from 'next/link'
import classes from './MainNavigation.module.css';
function MainNavigation(){
  return (
  <header className={classes.header}>
    <div className={classes.logo}>Awiland_MarketPlace</div>
    <nav>
      <ul>
        <li>
          <Link href='/'>AwilandHome</Link>
        </li>
        <li>
          <Link href='/popular'>Popular</Link>
        </li>
        <li>
          <Link href='products'>Products</Link>
        </li>
      </ul>
    </nav>
  </header>
  );
}
export default MainNavigation;
```

The ui(User Interface) component, contains JavaScript file called Card.js, which is a place holder to display products on the application. Code below:

```
import classes from './Card.module.css';


function Card(props) {
  return <div
className={classes.card}>{props.children}</div>;
}


export default Card;
```

Node Modules: This is a folder containing cache libraries, that are used by external modules that my project depends upon. Every time I do npm install, I am downloading the node_modules.

Pages:

I created dynamic pages when I moved my app into NextJs, meaning I created a folder and the file inside is an index.js JavaScript file that returns an asynchronous getStaticProps with the element inside being a context contains all the data for the created product. Below is the code:

```
import { MongoClient } from 'mongodb';
import ProductDetails from '../../components/awistore/ProductDetails';
function DetailsPage() {
  return (
    <ProductDetails image='https://www.thecarconnection.com/overview/mercedes-
benz_gl-class_2015#image=100498694'
      productType='Cars'
      address='City'
      price='#54,000000'
      description='My First Product'
    />
  );
}export async function getStaticPaths() {
  const client = await
MongoClient.connect('mongodb+srv://Ibro4realx:Mezozoyozi5@cluster0.firp8.mongodb.
net/Products?retryWrites=true&w=majority');
  const db = client.db();

  const productCollection = db.collection('products');
  const products = await productCollection.find({}, { _id: 1 }).toArray();
  client.close();
  return {
    fallback: false,
    paths: products.map(products => ({ params: { productId: products._id.toString() }, })),
  };
}export async function getStaticProps(context) {
  //fetch data from an API
  const productId = context.params.productId;
  console.log(productId);
  return {
    props: {
      products: {
        id: 'p1',
        image: 'https://www.thecarconnection.com/overview/mercedes-benz_gl-
class_2015#image=100498694',
        productType: 'Cars',
        address: 'City',
        price: '#54,000000',
        description: 'My First Product',
      },
    },
    revalidate: 3600,
  };
}export default DetailsPage;
```

API: API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software. [11]

This connects my app through the MongoClient library, to share my data with external users. It uses an asynchronous function handler to perform a request and result task. Requesting data from the client to create a new product, and sending results requested to the app. The code below:

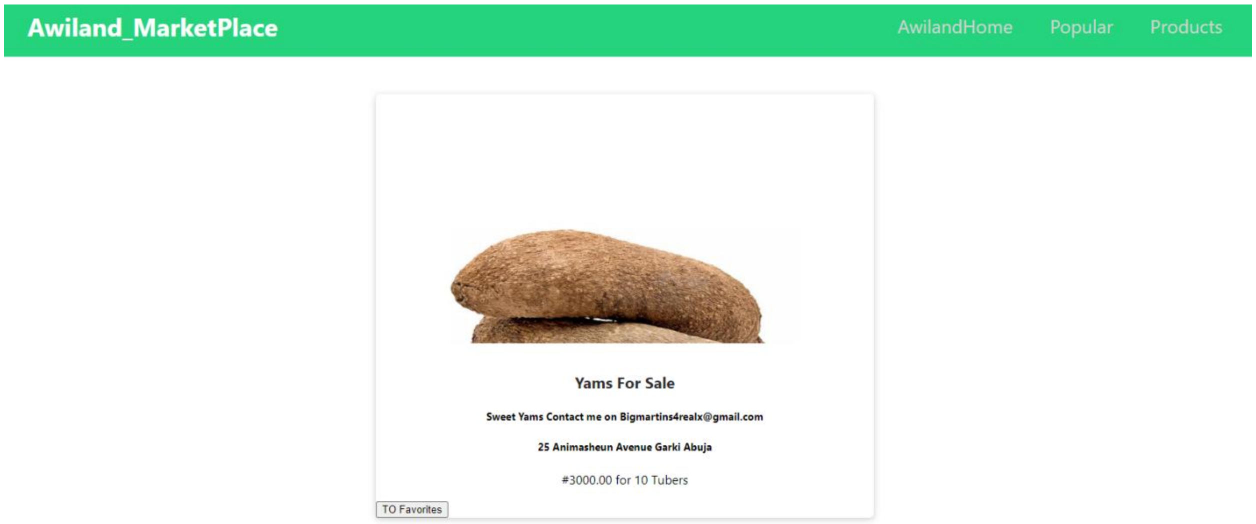
```
import { MongoClient } from 'mongodb';


//POST /api/ products
///api/ product

async function handler(req, res) {
  if (req.method === 'POST') {
    const data = req.body;

    const client = await
MongoClient.connect('mongodb+srv://Ibro4realx:Mezozoyozi5@cluster0.firp8.mongodb.net/Pr
oducts?retryWrites=true&w=majority');
    const db = client.db();

    const productCollection = db.collection('products');

    const result = await productCollection.insertOne(data);
    console.log(result);

    client.close();

    res.status(201).json({ message: 'Product created!' });
  }

}

export default handler;
```

8 Ethics

Staying focused, motivated, and having strong work ethic is an important part of being a successful Software Engineer. Work ethic which I can describe as a set of values based on the ideals of discipline and hard work. Building a strong work ethic will allow a person to discipline themselves so that hard work is almost automatic. Forming good habits such as focusing, staying motivated, finishing tasks immediately, and more helps to create a good work ethic that will create productivity.

9 Conclusion

In this report, I have presented a detailed explanation of my motivation and inspirations in developing AwilandMarketPlace.com. The goal was for users to be able to create a product in the app and post it, so that their customers can view and get product detail through the online store.



Yams For Sale

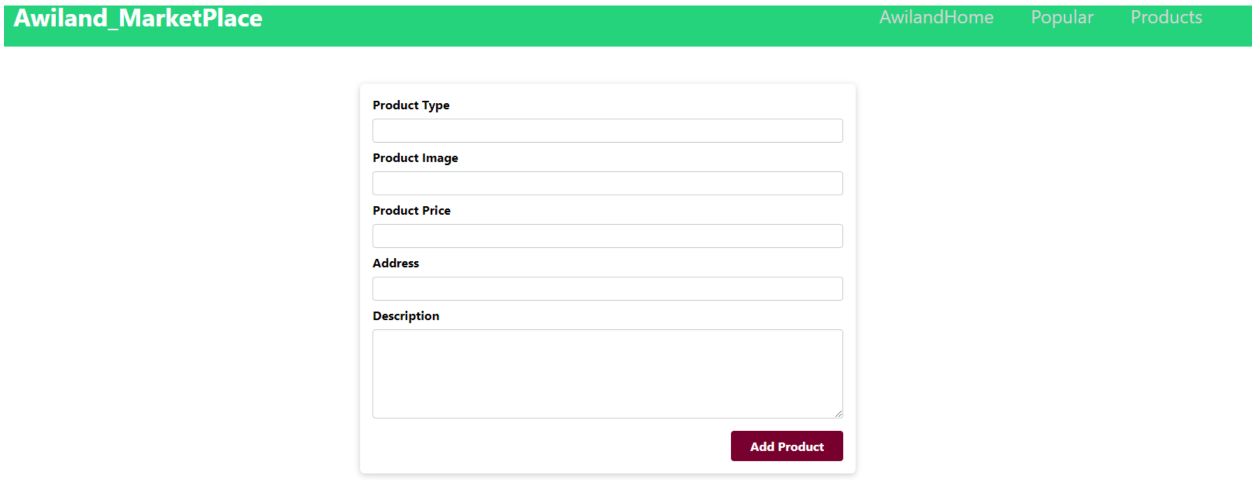
Sweet Yams Contact me on Bigmartins4realx@gmail.com

25 Animasheun Avenue Garki Abuja

#3000.00 for 10 Tubers

Figure 6 HomePage Result

Figure 6: Showing result of a product posted on the Awiland Marketplace.



Product Type

Product Image

Product Price

Address

Description

Figure 7 Products Page Result

Figure 7: Showing result of the form used in creating a post.

The dream is to use the understanding and knowledge I have developed during this project to fully develop this app for the purpose it was inspired. To improve the app, a user profile page could be added, so each product is linked to the product creator's page.

And the database could be done properly like creating a page that sorts product according to its category, type and date created.

10 References

[1] D. Taylor, "https://www.guru99.com/what-is-mongodb.html," 2022. [Online]. Available: https://www.guru99.com/what-is-mongodb.html.

[2] "TutorialsPoint," [Online]. Available: https://www.tutorialspoint.com/expressjs/expressjs_overview.htm#:~:text=ExpressJS%20is%20a%20web%20application,level%20protocols%2C%20processes%2C%20etc..

[3] "BlogRocket," [Online]. Available: https://blog.logrocket.com/creating-website-next-js-react/#:~:text=in%20October%202021.-,Next.,creating%20production%2Dready%20React%20applications..

[4] "TechTarget," [Online]. Available: https://www.techtarget.com/whatis/definition/Nodejs#:~:text=James%20Denman-,Node.,feeds%20and%20web%20push%20notifications..

[5] M. Barasa. [Online]. Available: https://www.section.io/engineering-education/how-to-implement-material-ui-in-react/.

[6] [Online]. Available: https://www.geeksforgeeks.org/how-to-use-card-component-in-reactjs/.

[7] S. Hayani. [Online]. Available: https://www.freecodecamp.org/news/an-introduction-to-next-js-for-everyone-507d2d90ab54/.

[8] [Online]. Available: https://reactjs.org/.

[9] [Online]. Available: https://reactjs.org/docs/react-dom.html.

[10] [Online]. Available: https://www.geeksforgeeks.org/what-is-react-router-dom/.

[11] [Online].

[12] [Online].

[13] H. Kinsley, "Reinforcement Learning," PythonProgramming, [Online]. Available: https://pythonprogramming.net/q-learning-reinforcement-learning-python-tutorial/. [Accessed 02 02 2021].

[14] MakeSigns, "Scientic Posters Tutorial," [Online]. Available:

https://www.makesigns.com/tutorials/scientific-poster-parts.aspx. [Accessed 09 02 2021].

[15] [Online].

[16] Arduino. [Online]. Available: https://www.arduino.cc/. [Accessed 09 02 2021].