PROJECT DOCUMENTATION

Project – Real Time Pizza Order Tracker

Made By – Shubham Kumar

Batch – TATA

----------I

I have made a Real-time pizza order tracker website. by using Java-script, Nodejs, express and have Mongo-db data base for this project.

The project will contain following features---

All the data will be stored in mongo DB database and will be retrieved from there only.

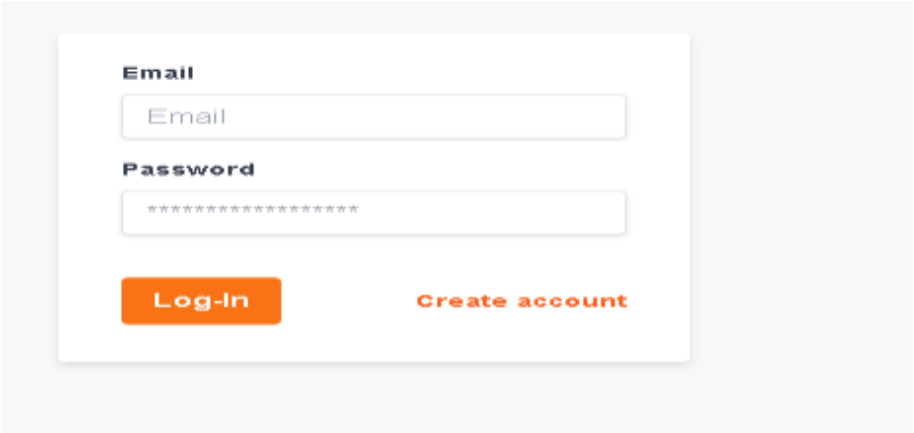
Features:

Project will have 2 types of user one will be the customer who can order pizza. And another will be the admin who will have access to all orders placed by the customer and admin will be responsible for fulfilling the the customers and updating the order details of the customers individually.

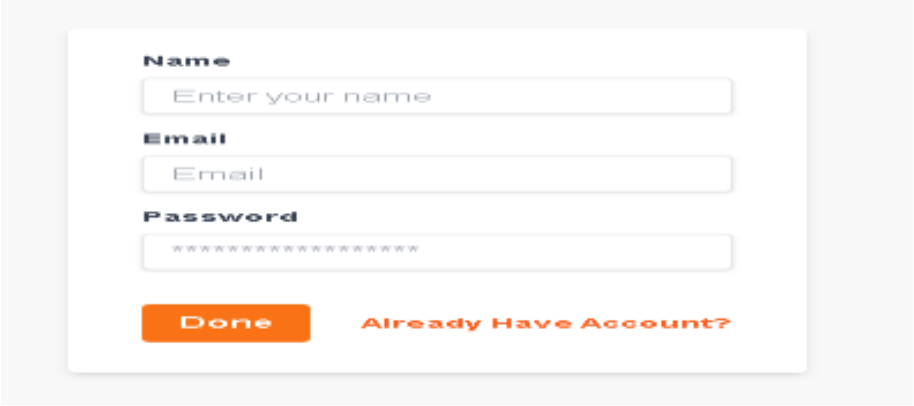
Customer Do’s

- Can login/register in the app.

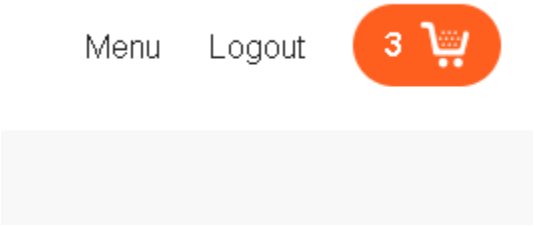
  Login



  Register

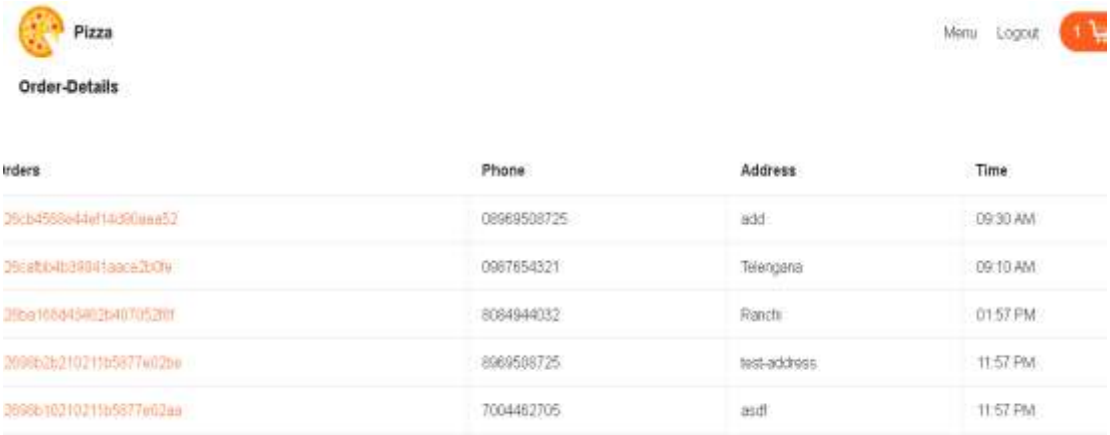


- Can Add items in the cart.



- Can Place order in the app.

- Can see their full order history.



- Can Track Their current Order status.

Customers Can't

- A customer cannot order pizza when he/she is not logged in.

- A customer cannot access admin order details page
- A customer cannot access other customer's order details.

Coding and implementation pages are shown below-

- A properMVC project structure has been followed for this project.

- All the static files are served inside the public folder.
- Tailwind CSS has been used for the Frontend styling part.
- I have also use EJS template engine for this project.

DATA BASE

- I am using Mongo-DB data base for storing the apps data.
- DB name – Pizza. Collections used for storing data are –

- **Orders –** Contains the details of all the orders placed by the customers with the customer Id and the unique order Id. Here is the example for the same-


```
  _id: ObjectId("62698b10210211b5877e02aa")
  customerId: ObjectId("62694361445aaea7c415afe6")
> items: Object
  phone: "7004462705"
  address: "asdf"
  paymentType: "COD"
  status: "Order-Placed"
  createdAt: 2022-04-27T18:27:28.718+00:00
  updatedAt: 2022-04-27T18:27:28.718+00:00
  __v: 0
```


- **Users-** Users contains the collection of all the users who are Registered in the app.


```
_id: ObjectId("626906d1c4a27aa707dae09e")
name: "Shubham kumar"
email: "abc@gmail.com"
password: "$2b$10$T4d6KTOVmolBjd0sPXg2HOT53ScF3uGlrofcM7bK8KLcgg7acIfeu"
role: "customer"
createdAt: 2022-04-27T09:03:13.029+00:00
updatedAt: 2022-04-27T09:03:13.029+00:00
__v: 0
```


- **Sessions –** Contains the session details

I have used Mongoose to create the Schema for storing the data and Mongoose schema is stored in a different file inside models-

1. Menus –

```
//will represent the collection in DB---
const mongoose = require('mongoose');

const Schema = mongoose.Schema

const menuScheme = new Schema({
    name: {
        type:String,
        required: true,
    },
    image: {
        type:String,
        required: true
    },
    price: {
        type:Number,
        required:true
    },
    size: {
        type:String,
        required: true
    }
});
const Menu = mongoose.model('Menu', menuSchema)

module.exports = Menu
```

2. Orders –

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

const orderSchema = new Schema({
    customerId: {
                type: mongoose.Schema.Types.ObjectId,
                ref: 'User',
                required: true
                },
    items: { type: Object, required: true },
    phone: { type: String, required: true},
    address: { type: String, required: true},
    paymentType: { type: String, default: 'COD'},
    paymentStatus: { type: Boolean, default: false },
    status: { type: String, default: 'order_placed'},
}, { timestamps: true })

module.exports = mongoose.model('Order', orderSchema)
```

3.User Schema-

```
const mongoose = require('mongoose');

const Schema = mongoose.Schema

const userScheme = new Schema({
    name: {
        type:String,
        required: true,
    },
    email: {
        type: String,
        required: true,
        unique: true
    },
    password: {
        type: String,
        required: true
    },
    role: {type: String, default: 'customer'}
}, {timestamps: true});
const User = mongoose.model('User', userScheme)

module.exports = User
```

Authorization

- **Authorization part is being handled inside the auth-controller.**

  >> I have used Bcrypt package for saving the encrypted password in the DB.

  >> I have used Passport package to authenticate the requests.

```
const LocalStrategy = require('passport-local').Strategy
const User = require('../models/user')
const bcrypt = require('bcrypt')

function init(passport) {
    passport.use(new LocalStrategy({ usernameField: 'email' }, async (email, password, done) => {
        // Login
        // check if email exists
        const user = await User.findOne({ email: email })
        if(!user) {
            return done(null, false, { message: 'No user with this email' })
        }

        bcrypt.compare(password, user.password).then(match => {
            if(match) {
                return done(null, user, { message: 'Logged in succesfully' })
            }
            return done(null, false, { message: 'Wrong username or password' })
        }).catch(err => {
            return done(null, false, { message: 'Something went wrong' })
        })
    }))
```

Routes

- All The Routes are imported inside the Routes folder in web.js.