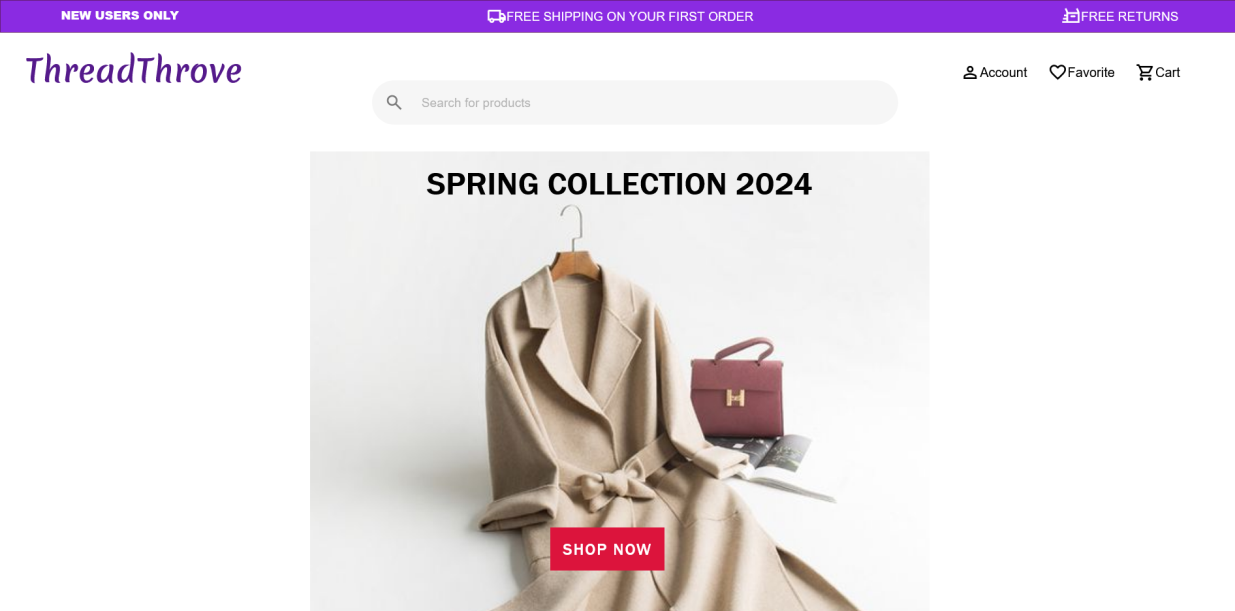
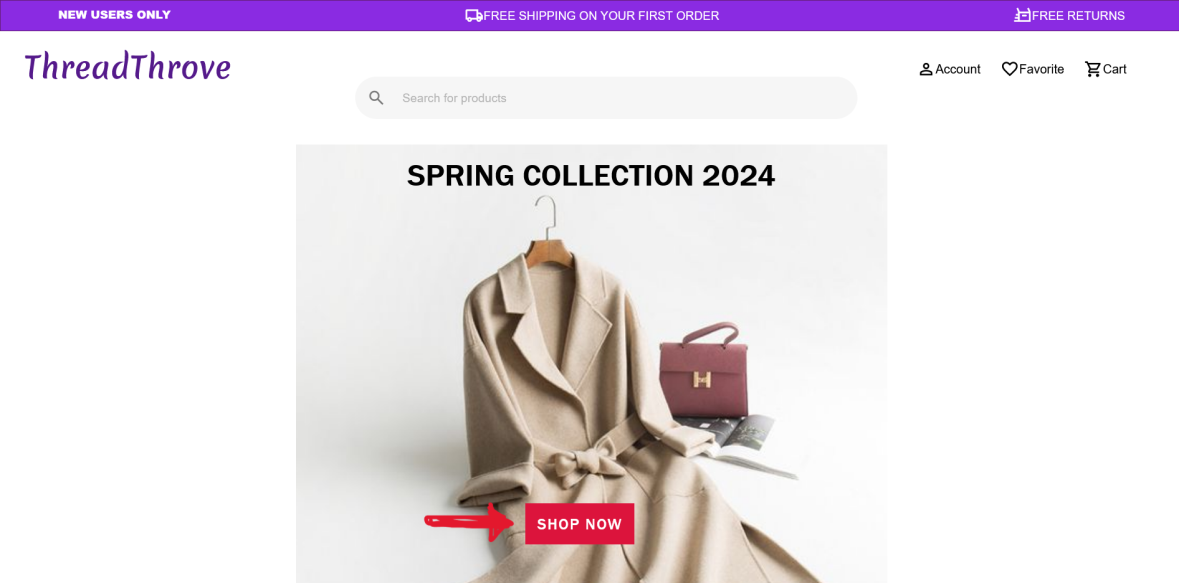
**CRUD OPERATIONS USING MONGODB (FrontEnd)**

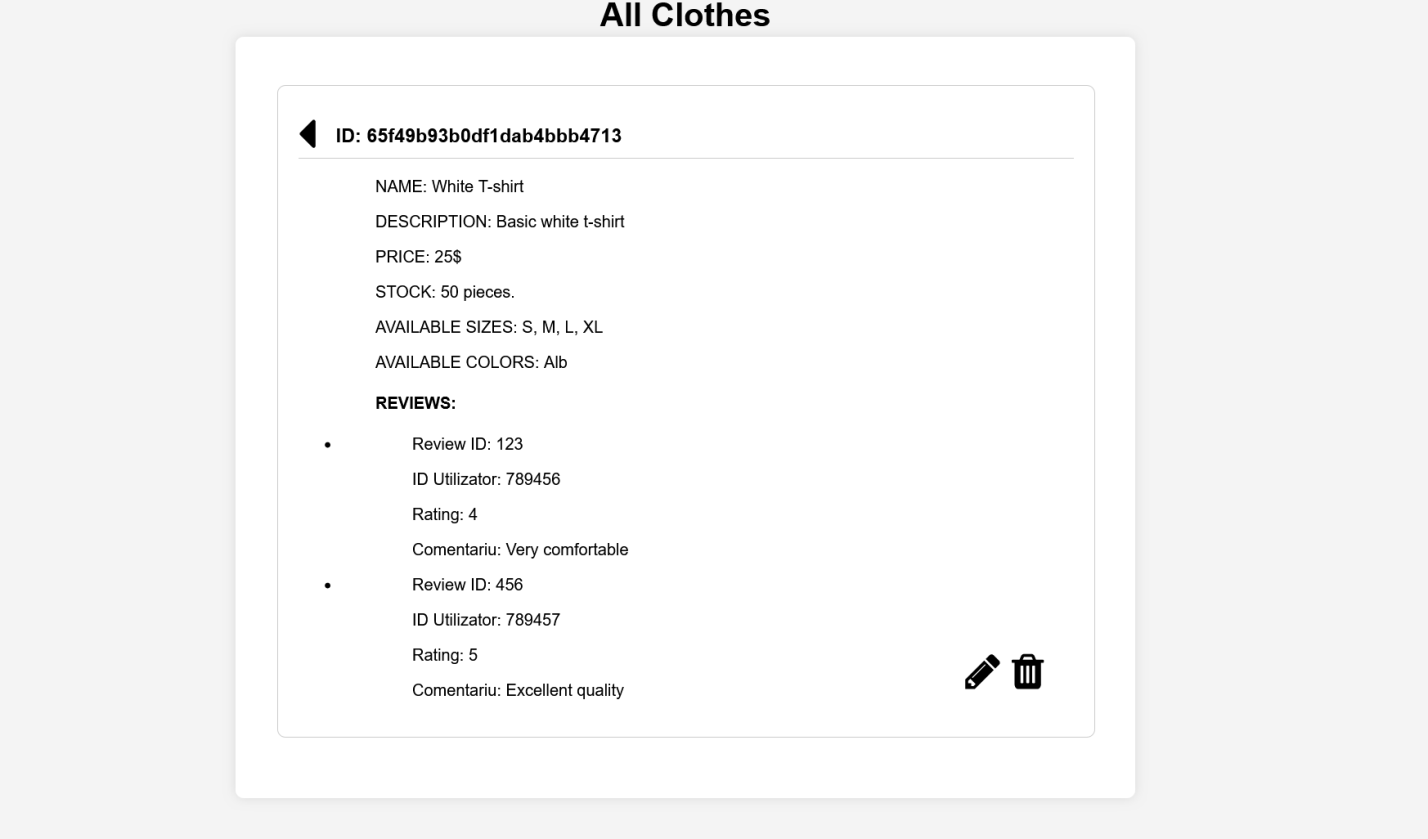
1. Download dependencies using npm install.
2. After dependencies are installed run the following command: nodemon app.js.
3. Type localhost: 5050, to open project in browser.
4. It should look something like this:



1. You can now click on **SHOP NOW** to see the clothes stored in mongoDB



1. To view more details about the clothes you can click on them, it will take you to a new page.



1. For updating the values of the clothes, press on the pencil icon.



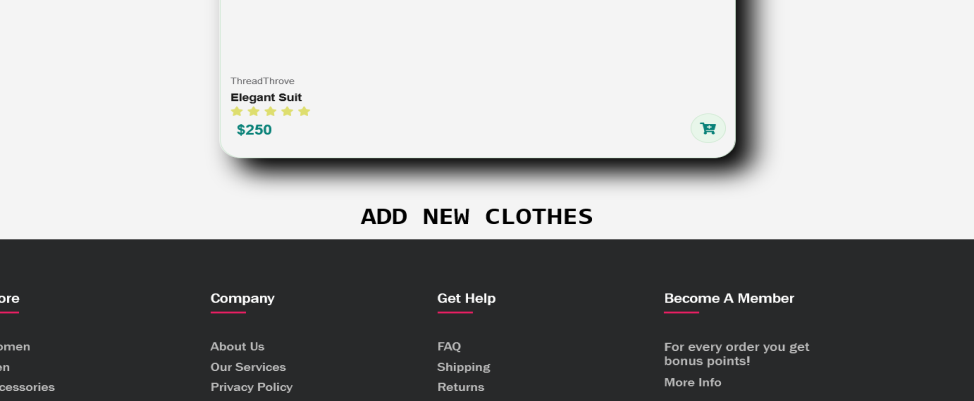
1. To delete item press on the trash icon.



1. To go back press on the arrow in the upper left corner.



1. To add new clothes scroll to the bottom of the page and press the button that says **ADD NEW CLOTHES.**



**CRUD OPERATIONS USING MONGODB (BackEnd)**

**App.js**

In app.js I connect to the mongo database and I also configure the express app.

I am using ejs as my view engine so I can incorporate JavaScript code, thus allowing dynamic display of data in pages. (I use it to display data from my database and also to reuse code for the header and footer)

I am also establishing my routes:

1. “/ ” – this is the default route. It displays the frontpage.
2. “/about” – this is an about page displaying information about the clothing store
3. “/women” –this is the women section. It displays women clothing
4. “/clothes” – this is the route where going to be focusing on, as the CRUD operations take place here

I am using app.use(“/clothes”, clothingRoutes) as a middleware for my routes that start with /clothes. When a request is made to a route that starts with /clothes, the Express app will use clothesRoutes to handle and process that request. (I used this so I could separe my routing from the main app, to make my code more organized)

**Model folder**

In the model folder I created the file clothing.js

This file contains the schema of my object, what values are required to be able to add a new clothing to my database.

**Routes folder**

The routes folder contains clothesRoutes.js

This is where I define the routes for my express Router. The next folder contains the functions that are implemented here.

**Controllers folder**

In this folder I declare my functions. Let’s see what all of them are doing:

clothes\_index -> this function shows all clothes ordered by creation date and time.

clothes\_details -> this function renders the details page (by id) that shows more information about the clothes the user has clicked on.

clothes\_create\_get -> this renders the form add-clothes.

clothes\_create\_post -> this function processes the form for adding a new garment, saves the garment to the database and redirects to the main page.

clothes\_delete -> this function deletes a garment (specified by ID) and redirects to the main page.

clothes\_update\_get -> this function displays the form for updating a garment (specified by ID).

clothes\_update\_post -> this function processes the form for updating a garment, updates the garment in the database and redirects to the updated garment details page.