**Task Description:**

We are a manufacturer. We are looking for a front-end web developer to build a website for us.

**Website Purpose:**

1. Your website has to be a tools manufacturer. The tool manufacturers could manufacture carpentry tools, masonry tools, electric tools, metalwork tools, or any tool. Some examples of tools could be a hammer, drill machine, screwdriver, wrench, Pliers, Measuring Tape, Chisel, Soldering Iron, shovel, electric saw, hacksaw, paint roller, paintbrush, sandpaper, nut, bolts, drill bit, clamp, workbench, nail gun, glue gun, etc.
2. Alternatively, your website could be a parts manufacturer. A part will produce parts of a machine or a device. Examples of parts manufacturers could be car parts manufacturers, Motor Bike parts manufacturers, Bicycle parts manufacturers, camera parts manufacturers, refrigerator parts manufacturers, air conditioner parts manufacturers, table fan parts manufacturers, Clock parts manufacturers, TV parts manufacturers, microwave oven parts manufacturers, computer parts manufacturer, etc.
3. Your website can not be typical e-commerce or any of your previous assignments. Make sure your design and website idea is unique. Make sure your code or design doesn't match any of the projects implemented in our course. Your website can not be related to your previous assignments or any practice project displayed in the course or our conceptual sessions.
4. Initially, do not spend too much time thinking about the design. Just spend 30minutes to an hour. And then finalize the idea. Start working on the Project. If needed, put a placeholder image and keep developing the functionalities. Once the functionality is done, come back to add proper images.

**Task Details:**

1. Home page will have a Header (simple navbar), Banner, Tools/Parts, Business Summary, Reviews, and Footer. The tools/parts will have either 3 or 6 tools/parts. Based on your design, you can put one or more tools/parts in a row. Each tool/part should have a relevant name, image, short description, minimum order quantity, available quantity, price (per unit price), a place order/purchase/book now/ buy now button. If a user clicks on the purchase or buy now button it will take the user to the purchase page.
2. Add two different sections on the home page and the 6 sections mentioned above. Tips: This is the place to shine. Add something different and unique to make your website looks different than others. And Make them relevant to your website. And make these two sections meaningful.
3. The business summary will show some summary of your business. There could be 3/4 summary on this summary section. For example, we served 100+ customers, 120M+ Annual revenue, 33K+ Reviews, 50+ tools, etc. Make sure you use some relevant icons here. You can use Hero Icons, React Icons, Font Awesome, Bootstrap Icons, Daisy UI or any icons that you like. To see an example, you will find an image in the images folder of this repository.
4. The `purchase' page will be a private/protected route. Please make sure that the private route redirects to the login page if the user is not logged in. After login, the user will be redirected to the page he/she wanted to go to. Also, after reloading the page of a private/protected route, the user is not redirected to the login page. The purchase page should display detailed information about the item the user has clicked somewhere at the top. It will also display the user's name and email address from the login system. The user will have a field to provide an address, phone number, and other necessary information (if applicable) to place the order or complete the purchase.

Please do not block the user if the email is not verified for our assignment evaluation purpose. Because it will stop the assignment evaluation if the email sending is not working, if you want, you can enforce this after receiving assignment marks.

1. On the Purchase page, users will be able to change the order quantity (increase/decrease) in an input field. The initial value of the quantity will be the minimum order quantity. However, the user won't be able to reduce the quantity below the minimum order quantity mentioned on the tool/part. Also, the order quantity can not be higher than the available quantity. You will display an error and disable the purchase button in both cases.
2. Implement an email/ password (login/Register) based login system. The registration form should have the name, and once a user is logged in, the user name and the logout button should appear on the header, which will log out the user once clicked. Implement at least one social login system (google, facebook, or github, etc.). Make sure to display login-related errors and password validation-related errors.
3. If a user is logged in, they will see another option on the header is called Dashboard. Inside the dashboard, a user (not an admin) will see options like My Orders, Add A Review, My Profile options on the side nav. This is the time to implement a nested route. Based on your website idea, you can change the name of these menu items. The My Profile link will be open for everyone. This means every user will be able to see my profile link and update their profile.
4. On My Orders page, the logged-in user will see only their orders. If the user wants, they should be able to cancel (canceling is just deleting the order) any order that is not paid yet. Ask for a confirmation message before canceling an order. Do not use browser default confirm. Instead, use a good-looking modal. Please note users will not see the cancel option for any paid order.
5. On the My Orders page, there will be a payment button for each order. The user has not paid yet. The pay button will take the user to the payment page. The user should be able to pay by using a credit card here. The payment page will display order details. Once the payment is completed for an order, the user will see the transaction id for that order on the my orders page.
6. On the Add A Review page, users should be able to add a review. A review will contain ratings ( a number 1 to 5 and a description). That review will appear on the home page reviews section. Right now, you will see every review on the home page. There is no limit on the number of reviews or the order of the review.
7. The My profile route will be available for every user (admin or non-admin. everyone will see this link). The user will see their name and email address on this profile in this link. Also, this page will have fields to add fields like education, location (city/district), phone number, LinkedIn profile link, etc. And users will be able to save this information in the database. Also, the user will be able to update this information.
8. If an admin logs in, they will not see the options that a user sees except My Profile. This means an admin will not see my orders and add a review link. Instead, an admin will see My Profile, Manage All Orders, Add A Product, Make Admin, Manage Products. Based on your website idea, you can change the name of these menu names. Manage products will be described in the bonus section.
9. An Admin should be able to make another user an admin. If an admin wants, they will be able to add a product on the add a product page. After adding the product, this product will appear on the home page. Please note, to add an image for a new product; you can just upload the image directly to the image hosting sites like imgbb, etc using API. Alternatively, you can have an input field to add the link to the image.
10. One route will be Blogs. This will be a open route (anyone visiting your website will see it). Please answer at least five questions on the blogs page. We recommend answering all six questions.

14.1 How will you improve the performance of a React Application?

14.2 What are the different ways to manage a state in a React application?

14.3 How does prototypical inheritance work?

14.4 Why you do not set the state directly in React. For example, if you have const [products, setProducts] = useState([]). Why you do not set products = [...] instead, you use the setProducts

14.5 You have an array of products. Each product has a name, price, description, etc. How will you implement a search to find products by name?

14.6 What is a unit test? Why should write unit tests?

1. Create My Portfolio route. In this route, you will have your name, email address, educational background, list of technologies or skills you have as a web developer. Add links of three of your projects (live website links). Adding descriptions or screenshots of your projects are optional.
2. Create a meaningful 404 page (not found page). Add a meaningful image on the 404 page.
3. Use a database to store information. It could be mongodb atlas or any other database.

**Bonus:**

1. Meaningful readme.md file on both the client and server sides containing your website name, a link to your live site. Add at least five bullet points that mention your website's different features and functionality.
2. At least 15 meaningful GitHub commits for the client-side and at least 10 meaningful commits for the server-side repository.
3. Make the website responsive. Make sure the site looks different on desktop and mobile responsive. Tablet responsive is optional.
4. Make the website meaningful and consistent in look and feel. Give your website a relevant name. Images and all the content of the website have to be appropriate. No lorem ipsum please.
5. Clean and organized Code. Organize components and add comments when needed. use .env file on the server, .env.local file on the client to hide db user and password, etc. Also use the .gitignore file
6. On the Manage All Orders page for the admin, the logged-in admin will see the orders placed by every user. If multiple users use this website to place orders, everyone's order will be displayed here. The orders that are not paid will show unpaid. And the orders that are paid, an Admin can update the status of a paid order. After placing an order and completing the payment, every order will have a default status: pending. On the Manage All Orders page, an admin will be able to update the status of the pending to shipped status. This could be a simple button to update the pending status. (don't overthink, this is a simple button to update the status field of the order. Use Id to find the order and set the status to approved). Also, the admin will be able to cancel (cancel means admin can delete an order) that is not paid yet. Make sure to add a confirmation before deleting.
7. Implement the basic version of the jwt token. Upon login, you will create a jwt token and store it on the client-side, and for the necessary pages, you will send the token with the call and verify the user. Implementing 401 and 403 is optional. Ensure you have implemented jwt token and create token and store it on the client-side for both email/password-based authentication and social login.
8. Similarly, you will need to protect the admin route and on the server side verify the admin level api requests.
9. on the Manage Products page, an admin can delete any products. Please make sure there is a confirmation. After deleting that product will not appear on the home page. Use a meaningful and good-looking confirmation modal. Avoid using browser default confirm.
10. Use react query for at least one API call.
11. Use react hook form at least one form. For the rest of the forms, you can either use react hook form or basic html form or any other form you want.

**Optional but highly encouraged to do:**

Make sure you have done everything on the main part and bonus. Deploy and test everything. After that, you may try optional things.

1. Add animation in the home page and other places.
2. We encourage you to use axios and implement interceptors to inject authorization headers in one place other than adding an authorization header for every call separately.
3. Send a confirmation email after placing an order. Even if the email doesn't get sent or goes to the spam folder. Also, it will be nice to send an email once an order is shipped to the user.
4. If possible, on the Manage All Orders implement a button to show all pending orders. Another button to show all shipped orders, all unpaid orders. Add search to find an order by email address of a user. The list of orders could be sorted by the newest one at first. Consider implementing pagination on this page.
5. Display order status on the My Orders page. So that a user can see the status of the order whether it is pending, or shipped.
6. Please Use icons whenever applicable and use fonts (google fonts)
7. Make the footer a little more realistic
8. Optimize your images
9. Add something extra of your own. This will help you in the future.
10. Also, if you want, you can use mongoose.
11. If possible, replace the default react favicon and give an icon for your website.
12. Display the latest (newest) 6 reviews on the home page. The most recent reviews added to your website.

**Additional information:**

1. You can use a local image or host image anywhere or both.
2. You are free to use any css library you want. But, we recommend using tailwind css. Also, you can use one or more component libraries based on your need.
3. Try to host your site on Firebase (Netlify host will need extra configuration) and deploy your server-side code to Heroku. Don't forget to replace localhost links on the client-side. Make sure your website is working without your computer running. Check your website on your phone after the final deployment.

**What to submit**

1. Client-side code github repository
2. Server-side code github repository
3. Live website link
4. For our testing purpose, you will need to provide admin credentials (an email address and a password)
5. [optional] Project and code overview demo video

**Some Guidelines:**

1. Getting Errors is part of becoming a web developer. So, ❤️ love the errors. They will test your patience and will make you a better developer 💪.
2. Do not waste much time on the website idea. Just spend 30 minutes deciding, finding a sample website, and working on it.
3. Do not waste much time finding the right image. You can always start with a simple idea. Make the website and then add different images.
4. Don't look at the overall task list. Just take one task at a time and do it. Once that is done, pick the next task. If you got stuck on one task, move on to the next task.
5. Stay calm, think and work sequentially. You will make it.
6. Don't wait for the last moment to deploy the server-side or client-side code. I will recommend deploying the server-side with the root API get URL. And then keep deploying to the Heroku server.
7. Keep incremental deploy to firebase as well.
8. Do not copy-paste code from anywhere. Write your code and put your best effort into this assignment.

Have FUN! Have Patience. You will make it. 🔥🔥🔥