Task Description:

You have to create a website to provide a quality learning platform in the domains of Computer Science/ Information Technology/Programming Languages/ Web technology/ anything else (anything related to education). For example, JavaScript tutorial. Please note that you will select one of the options mentioned above/ any one topic regarding education. You can't make a website with a JavaScript tutorial, as we have given an example on it.

1. Make sure your design and website idea are unique. First, finalize your idea (What type of website do you want to build). Then google the site design or visit ThemeForest to get your website idea. However, your website can not relate to your previous assignments or any demo project displayed in the course or our conceptual sessions.

1. Give your website a name. The name should appear on the website and be displayed on the website's title.

1. It will contain a navbar with the options: website logo, website name, Courses, FAQ, Blog, toggle theme (dark/ light), and User Profile Picture (or login). The user profile picture on the navbar is conditional. If the user is signed in, the navbar will show the profile pic. The user's name will be visible when the mouse is in the picture.

1. After clicking "Log in," it will redirect to a login page. 3 types of login systems you have to implement: Email & password login, Google, and GitHub login. The login page will have a toggle option to go to the register page.

1. In your Email and Password registration form, there should be an input box for "Full name," where the user will type their full name. Other fields will be Photo URL, Email, and Password.

1. Clicking "Courses" in the navbar (You can change this name to any meaningful name according to your design, but the purpose of this button should be the same according to our description) will redirect to a page that shows the courses. (Creating a fake dataset is necessary here). You have to create at least 6 courses (fake data). Design & course information is all up to you. (Load data from the server side - must). You can create one, two, or more courses in a row.

1. The Courses page will be divided into two parts: a sidebar (either on the left side or on the right side according to your design) & content detail section. In the sidebar, there will be at least 6 options (course options or course categories, or anything else). Every option is clickable. (You will need to create necessary fake data)

1. When you click an option (course card or an option on the left/right sidebar), It will take you to the course detail page. Content details depend on you. But, it must include a header, introductory description/course body, and one image. All details must be relevant to the topic. You may include a bullet list/ table/ chart if needed.

1. The course/content detail section will have a heading at the top of the page. The heading will have an icon/ button on the side of the heading. When you click on that button, it will create a pdf and will be downloaded (This will give you more fun. Try out this after completing all of your tasks. Hint: Explore react-to-pdf package). Your pdf should contain some relevant information about the course details.

1. There will be a button at the bottom of the content detail called "Get premium access." Clicking the button will take the user to the Checkout route. This route will be a private/protected route. Please ensure 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 checkout page.

1. Your checkout page will be a dynamic route with route parameters having the id of the content that the user had clicked. On the checkout page, you will display the name based on the id.

1. Use the Environment variable to hide the firebase key.

1. If you reload the protected/private route (after login), this page will not redirect the user to the login page. Instead, it will keep the logged-in user on the protected route.

Bonus:
1. Minimum 12 meaningful git commits on the client-side repository and minimum of 5 meaningful commits on the server-side repository.

1. Make your website mobile & desktop responsive (tablet responsive is optional).

1. Display errors when the user's email address or password doesn't match.

1. Clicking "Blog" will redirect to a public route. That route will have 4 questions & answers.

    4.1 what is `cors`?

    4.2 Why are you using `firebase`? What other options do you have to implement authentication?

    4.3 How does the private route work?

    4.4 What is Node? How does Node work?

1. (client-side repo) Meaningful readme.md file containing your website name and link to your live site. And at least five bullet points mentioning your website's different features and functionality. And a list of technologies (frameworks, libraries) used in your project. (added a meaningful readme.md file for the server-side repo is optional)

1. Create a 404 page.

1. Make sure your site looks reasonable. The design and color selection is meaningful.

2. Clean and organized Code (folder structure). Organize components with meaningful names, and add comments when needed.

Optional (But Highly Recommended):

1. Add a link for Password reset (use toast/ alert), (Don't worry if the email goes to the spam folder.)

3. Add "active route" to indicate the route which you are visiting.

1. Use react-hook-form package to make forms on your website

1. Explore & use these react packages in your website : react-icons, react-image-magnify, react-awesome-slider, react-loader etc.

1. Feel free to add some relevant FAQ in your FAQ route

1. If you have sometime, try to implement dark/light theme toggle.

1. Optional but highly recommended: A detailed, realistic, complex & stylish footer.
2. When a user clicks on their profile image, it will redirect to the user details page. This page will show the user's detailed information (name, email & profile picture). The logged-in user can update their profile(image, displayName) by clicking on a button. (Alternatively, The profile edit form could be in a modal)
3. Add something extra of your own. This will help you in the future to differentiate your project from others.
4. If you want to take a challenge. Consider using `react-leaflet` anywhere in your application

Additional information:

1. You can use a local image or host image anywhere or both.
2. You are free to use any CSS library (bootstrap, tailwind) you want. if you want, you can use both bootstrap and react-bootstrap. Also, if you want, you can use any tailwind component library such as DaisyUI, etc.
3. Try to host your site on Firebase (Netlify hosting will need some extra configurations)
4. Host your server-side application on Vercel.
5. If you have any issues with hosting or Github push, please join the "Github and deploy" related support session.

What to submit:

1. Your client-side code github repository
2. Your server-side code github repository
3. Your live website link

Some Guidelines:

1. Do not waste much time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
2. 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.
3. Don't look at the overall task list. Just take one task at a time and do it. Once it's done, pick the next task. If you get stuck on a particular task, move on to the next task.
4. Stay calm, think before coding, and work sequentially. You will make it.
5. Be strategic about electricity issue.

No Pain, No gain:
`Have Patience! Work Hard! YOu will surprise yourself, and eventually, you will be proud of yourself.`