Assignment_ID: assignment_category_0010

Volunteer Management Website

Calling all Web Wizards!

We're seeking a bright and curious Junior MERN Stack Developer to join our awesome team. You'll get to work side-by-side with our expert developers on cutting-edge projects using hot technologies like MongoDB (databases), Express.js (backend), React.js (frontend), and Node.js (runtime environment). This is your chance to level up your skills and be a part of building innovative web applications that make a real impact.

Read this document carefully and try to fulfil all the requirements.
Make your website stunning!

- **Easy on the Eyes:** Use color combinations that are pleasant to look at and avoid harsh contrasts.
- **Keep it Clean:** Move away from a cluttered "gobindo" design (think lots of elements scattered together).
- **Organized & Balanced:** Ensure all elements on the website are properly aligned and have enough space.
- **Component Customization:** Even if you're using pre-built components from libraries like Daisy UI, don't be afraid to customize them! Change the styling to fit your website's unique look and feel, instead of just copying and pasting the default design.

Your website can not be related to your previous assignments' layout/ design or any practice project shown in the course modules or our conceptual sessions.
Ex: You can't copy any design or similar functionality/ layout of-

- Dragon news
- Espresso Emporium
- Genius Car
- Career Hub Website,
- Any conceptual session projects
- Any of your previous assignments

If any similarities are found, you will get zero(0) as a penalty

Website Themes: (Volunteer Management Website)

You need to create on building a user-friendly platform for volunteer management where a user can create a volunteer need post, update and delete a post, a user can be a volunteer for others post.

Key Rules:

- Include a minimum of 18 notable GitHub commits on the client side
- Include a minimum of 8 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website and a live site URL. Include a minimum of five bullet points to feature your website.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views.
- After reloading the page of a private route, the user should not be redirected to the login page.
- Use the Environment variable to hide the Firebase config keys and Mongodb credentials.
- Don't use any Lorem ipsum text; you can not use the default alert to show any error or success message.

Main Requirements

1. **Navbar:** The navbar will contain the
    - Website name/logo,
    - Home,
    - Need Volunteer Page,
    - Conditional "Login" and "Logout"
    - My Profile

a. The “Login” button is conditional, if the user is not logged in it will show the “Login” Button. If a user clicks on “Login” it will redirect to the login page.
b. But If the user is logged in here you will show the user photoURL, when you hover over the image it will show the displayName. And it will show the “Log out” button.
c. On clicking on My Profile it will show a dropdown menu. It will show the following routes -
    i. Add Volunteer Post
    ii. Manage My Post

🎯**Except Home, Need Volunteer Page and login/register all the routes will be private/protected routes.**

2. **Layout:**
    - Create a meaningful Navbar according to the above info.
    - **Footer:** Create a reasonable and meaningful footer. (including website name, copyright, contact information, social media links)
    - **404 page:** Add a 404 page/Not Found Page

🎯 **Make sure to keep the navbar and footer on all the pages except the 404 page.**

3. **Login Page:** When you click the login button on the navbar it redirects to the login page. You have to use a password and email-based authentication to log in. The login page will have-
    a. Email
    b. Password
    c. Google login/ GitHub- implement any of one
    d. A link that will redirect to the Register page

    🎯**Here the email and password should match with the registered email and password. If it doesn’t match, show an error message. You can show an error by using toast/sweet alert if you want.**

4. **Register Page:** You have to use a password and email-based authentication to register. The Register page will have the following -
    a. Name
    b. Email
    c. photoURL
    d. password
    e. A Link that will redirect to the login page
    ★ For password verification you need to follow this -
        - Must have an Uppercase letter in the password
        - Must have a Lowercase letter in the password
        - Length must be at least 6 character
    ★ If any of this isn't fulfilled it will show an error message /toast
    ★ After successful login or Register you need to show toast/sweet alert

🎯**Don't implement email verification or forget password method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.**

Home Page

5. **Banner/Slider:** Add a slider (you can use any type of slider/carousel) with a minimum of 3 slides and meaningful information. Make sure the banner/slider seems eye-catching.

6. **Volunteer Needs Now Section:** This section will show a maximum of 6 cards which will showcase volunteer need posts with upcoming deadlines. Each card will show the following info-
    - Thumbnail
    - Post Title
    - Category
    - Deadline

- View Details Button

🎯To show the posts according to the upcoming deadline you can use the MongoDB sort method.

➔ **See all button:** Below the 6 cards, there will be a see all button that will redirect the user to the Need Volunteer Page.

7. **Extra Section:** Add 2 relevant and meaningful extra sections on the Home page.

Add Volunteer Post Page

8. **Add Volunteer Post:** Create an Add Volunteer Post page where there will be a form for the user to add a post showing the need for volunteers. The form will have:
    a. Thumbnail
    b. Post Title
    c. Description
    d. Category => for example- healthcare, education, social service, animal welfare.
    e. Location
    f. No. of volunteers needed
    g. Deadline => Take the date using React datepicker
    h. Organizer name and organizer email (Logged-in user email & name) (read-only)
    i. Add Post Button

    This will be a private/protected route.
    When you fill in the data and submit the “Add” button, these data will be stored in your database and you will show a success message through toast/sweet alert.

Volunteer Need Post Details Page

9. **Volunteer Need Post Details Page:** The volunteer need post detail route will be a private/protected route. Please make sure that if the user is not logged in, the private route redirects to the login page. On this page, you will show all the information you have stored in the database and there will be a “Be a Volunteer” button in this page.

Be a Volunteer page/Modal

- On clicking on this “Be a Volunteer” button it will take a new route/open a modal. Which will have a form containing these input-
    - Thumbnail (read-only)
    - Post Title (read-only)
    - Description (read-only)
    - Category (read-only)
    - Location (read-only)
    - No. of volunteers needed (read-only)
    - Deadline (read-only)
    - Organizer name and organizer email (read-only)

🎯**All the above data will show from the database and will be set as the default value.**

- Volunteer name and volunteer email (Logged-in user email & name) (read-only)
- Suggestion (editable)
- Status => By default it will be “requested”
- “Request” Button

Clicking on the request button will store all the information in a new collection in the database. When a user requests here it will decrease the “No. of volunteers needed” amount. You can do this by using the $inc operator in MongoDB.

Need Volunteer Page

10. **Need Volunteer Page:** On this page, you will show All the Volunteer Need posts users have added. You need to show it in a card format. Each card will contain-
    a. 3-4 information from that post

b. View Details button – will redirect to the **Volunteer Need Post Details Page**

- **Search Functionality:** On the top of this page, you need to implement search functionality through a search input based on the Post Title. You can implement it through the backend.

Manage My Post Page

It will be a private/protected route. On this page, a user can see all the Add Volunteer Posts he/she has added to the database. here a user can only see his/her added data, but he/she can not access other's data. On this page, there will 2 different sections.

11. **My Need Volunteer Post:** Here you will see all the posts you have added through the "Add Volunteer Post Page". You need to show the data in the table layout. Each row will contain -
    a. 2-3 information about that post
    b. Update Button
    c. Delete Button

🎯If there is no data for My Need Volunteer Post then show a meaningful message or something relevant.

12. **Update Page:** When a user clicks on the "Update Button" it will take the user to the Update page, where there will be a form for the user to update a Need volunteer Post. All the previous data will show as the default value. The form will have the following:
    - Thumbnail
    - Post Title
    - Description
    - Category => for example- healthcare, education, social service, animal welfare.
    - Location
    - No. of volunteers needed
    - Deadline => Take the date using React datepicker

- Organizer name and organizer email (Logged-in user email & name) (read-only)
- Update Post Button

This will be a private/protected route. When you fill in the data and submit the "Update" button, these data will be updated to the previous data in your database and you will show a success message through toast/sweet alert.

🎯🎯**(Optional):** If you don't want to create an Update page, you can also use a **modal** to update your data. For this when you click on the "Update" button it will open a modal but make sure you are logged in before updating the data.

🎯**For all the CRUD operations, show relevant toast/ notification/ sweet alert with a meaningful message**

13. **Delete Button**- If the user clicks the delete button, the Need Volunteer Post will be removed from the data. Before the delete, ask for a delete confirmation.

14. **My Volunteer Request Post:** Here you will see all the posts you have added through the "Be a Volunteer Page/Modal". You need to show the data in the table layout. Each row will contain -
    a. 3-4 information
    b. Cancel Button
- If the user clicks the Cancel button, the Be a Volunteer Request will be removed from the data. Before the cancel, ask for a cancellation confirmation.
- If there is no data for my volunteer request post then show a meaningful message or something relevant.

15. **Dynamic Title:** Make your website title Dynamic. For every Route change, The Website Title will be changed based on that route.

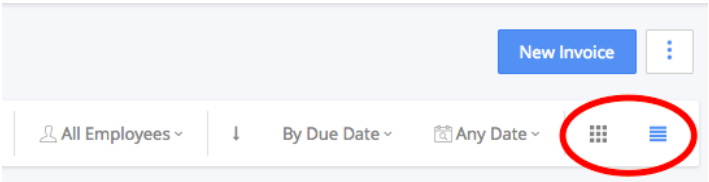
Challenges Requirements:

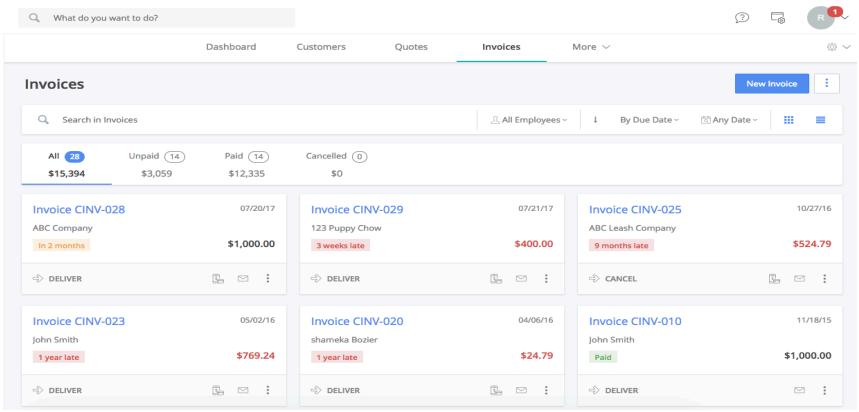
1. On the Need Volunteer Page if the No. of volunteers needed is 0 then it will show a message and the user can not add it through “Be a Volunteer”.

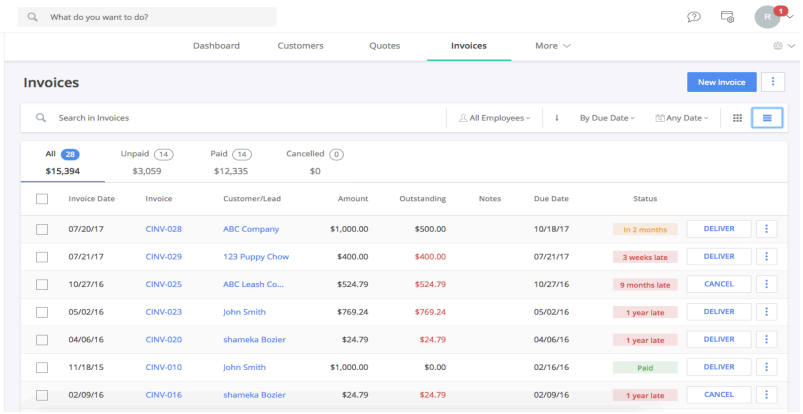
2. **Dark / Light Theme:** Implement a dark / Light theme toggling in your Navbar for the whole web application.

3. **JWT Authentication:** Upon login, you will create a JWT token and store it on the client side. You will send the token with the call and verify the user. Implementing 401 and 403 is optional. Ensure you have implemented the JWT token, create a token, and store it on the client side for both email/password-based authentication and social login. You must implement JWT on your private routes.

4. **Change Layout:** In Need Volunteer Page, Initially there will be a three-column layout. On clicking the change layout button, the layout will change in the card to a table form layout.







Optional (But Highly Recommended):

Implement any two tasks from the following optional list:

1. Try to use any other tailwind CSS library like - mamba Ui, shadcn, chakra UI, flowbite.
2. Add a spinner when the data is in a loading state. You can add a gif/jpg, use any package or customize it using CSS.
3. Explore and implement any of the animations from the Framer Motion.
4. Add one extra feature of your own. This will help you in the future to differentiate your project from others.
5. Implement Pagination in the Need Volunteer Page. Show 6-9 services per page.

What to submit:

1. Your Assignment ID/Variant
2. Your client-side code GitHub repository
3. Your server-side code GitHub repository
4. Your live website link

Additional information:

1. You can host images anywhere.
2. You can use vanilla CSS or any library.
3. Try to host your site on Firebase (Netlify hosting will need some extra configurations)

Firebase Hosting Setup Complete Issue

4. Host your server-side application on Vercel. If needed, you can host somewhere else as well.
   How to deploy a Node/Express server using Vercel CLI
   Some Common Vercel Errors
5. Make Sure you deploy server-side and client-side on the first day. If you have any issues with hosting or GitHub push, please join the "Github and deploy" related support session.

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 the electricity issue.

6. use chatGPT to generate JSON data. You can use chatGPT for other purposes as well.