

Fitness Tracker Website

assignment12_category_0009

Seeking a Full-Stack Developer for an Innovative Fitness Tracker Project.

Our client is seeking a passionate and skilled Full-Stack Developer to join their team and play a pivotal role in developing a cutting-edge fitness tracker. The ideal candidate will possess expertise in both front-end and back-end development.

Make sure your website design is unique. Visit [ThemeForest](#), [Dribble](#), [Behance](#), etc. to get some ideas. You can explore component libraries other than DaisyUI. Remember, a unique project will add more value to your portfolio.

🚩: 0 [changes will be tracked here]

Main Requirements

1. Focus on making the website visually appealing. Ensure that
 - Color contrast is pleasing to the eye
 - The website does not have a gobindo design
 - The website has proper alignment and space
 - If needed, customize the design of any component you are taking from any component library. (For example, you are using Daisy UI & have taken a card component from Daisy, if needed, customize the styling of the card to make it reasonable rather than just copy & paste it.)

Note: 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

- **Career hub**
 - **Dragon news**
 - **Coffee store**
 - **Ema john**
 - **Car Doctor**
 - **Bistro boss**
 - Any of your previous assignments or conceptual session projects. If any similarities are found, you will get **zero(0)** as a penalty.
2. Make sure to keep the **navbar** and **footer** on all the pages except on the **404 page**. Create a reasonable and meaningful footer. (including website **logo, name, copyright**, some contact information, social media links, address, etc.)
 3. For all of your CRUD operations and login/register, you have to show a toast/alert (do not use browser alert).

Navbar

Your website should have a navbar with the following information:

1. Website name/logo.
2. Home page.
3. Gallery page.
4. Trainer Page.
5. Classes page.
6. Dashboard.
7. Community/Forums page.
8. Conditional login/register.
9. User Profile.

Note: The User profile picture on the navbar is conditional based on login. If the user is logged in, the navbar will show the profile picture and logout button; otherwise, it will show the Login button.

Homepage 🏠

👉 **Banner section:** A slider/banner/ a meaningful section. Inside the banner, there will be a Heading Title, a Short Description, and a button that will redirect the user to the **classes** page.

👉 **Featured section:** Show some cards about website features.

💡 For ideas visit this but don't copy this design 🔗 [link](#)

👉 **About section:** this section will have some info about the organization.

You can use any type of layout or design but try to match it with your whole website design.

👉 **Featured classes:** show some classes Minimum 6. [see optional]

👉 **Testimonials or Success Stories:** Quotes or stories from satisfied users or clients showcasing their achievements using the platform.

👉 **Blog or Latest Articles:** Highlights of recent Community/Forums posts or articles related to fitness, nutrition, and exercise. Links to read more or explore the blog section.

👉 **Newsletter section:** Create a section that will have a name, email field, and a subscribe now button. Clicking the subscribe now button will save info in the database.

🔒 Remember: Users are not required to log in for newsletter subscriptions.

👉 **Team section:** Show a minimum 3 Coach/trainer profiles along with some info.

Gallery Page 🖼️

👉 **Page Title:** The background and the center of the section will have the page name.

👉 **Gallery section:** On this page, you have to implement infinite image scrolling. Initially, there will be 12 grid items. After scrolling down the rest of the image will be shown. Also, add simple animation when the image loads. **(you can use both database or local images)**

💡 You can explore this video: 🔗 [link](#)

Trainer Page 🏃

👉 **Trainer Section:** Show all the Trainer profiles. Each Card contains the following info.

- Trainer name
- Profile Image
- Years of Experience
- Social icons
- Available time slot.
- **Know more**

📝 Clicking the **Know More** button will redirect to the Trainer details page. Here you have to show the available slot of that trainer.

🕒 Each slot will be 1 hour in length. **Ex:** If a Trainer's available time is **6** hours a day then there will be 6 slots.

👉 **Be a Trainer Section:** below there will be a section where the CTA button is called **Become a Trainer**. Clicking the button will redirect the user **Be a Trainer page**.

Be a Trainer Page (Private)

👉 This page will have a form where the user can submit the following info:

- Full Name
- Email (**read-only**)
- Age
- Profile Image
- Skills (use checkbox for selecting multiple skills)
- Available Time in a week
- Available time in a day.
- Other info depending on your project.
- Applied Button.

📝 On clicking the apply button all the info will be stored in the database.

Trainer Booked Page 🏠 (Private)

👉 From the **Trainer Section** Clicking the available slot will redirect the user-booked page where you will see 3 types of plans. (ex: silver, gold,

diamond). Each package will include some classes and other facilities he/she can get. There will be a **join now** button below each plan.

 clicking on the **Join Now** button all the info will be stored in the database. [**recommended an optional feature**]


✨ **Optional method[Highly recommended]:**

👉 clicking on the **Join Now** button will redirect the user to the payment page.

Payment Page **(Private)[optional]**

👉 On this page, you have some info:

- ❖ Trainer name (**read-only**)
- ❖ Slot that you want to book.
- ❖ Package name
- ❖ Price
- ❖ Your name, email & other info (**read-only**)
- ❖ **Confirm button**

 Clicking on the confirm button will save all the information in the database.

Classes Page

👉 **Weekly schedule Section:** in this section, you have to implement a weekly schedule where the user can see daily activities.

 You can take inspiration from [this](#). [For more info see the optional part]

👉 **All Classes Section:** Here you will see all classes. [implementing pagination is optional]. Clicking on each class will show details info about that particular class. Also, there will be a join now button that will redirect the user to the Trainer page.

 Only a Trainer can add new classes.

Forum page **B**

👉 On this page, you have to show a minimum of 6 posts. Below the six posts implement a pagination using the backend.

👉 There will be an up-vote and down-vote system. If the user logged in he/she can upvote/down-vote a forum.

💡 hints: you can explore Facebook's like and dislike system

Dashboard **(Private)**

👉 In the dashboard, the pages will be conditional based on user role.

Admin **(Private)**

An admin will see The following routes/pages 👉

- ❖ **All subscribers:** here you have to show All newsletter subscribers in tabular format.
- ❖ **All Trainers:** On this page, you have to show all Trainers in a tabular format.
 - 👉 There will be a **conditional** pay button. If the admin clicks the pay button it will redirect to the payment page where the admin can pay the trainer's monthly salary.
 - 💡 hints: you have to count months from a trainer Joined.
 - 👉 After confirmation of successful payment redirect the user to the **All Trainer** page and payment status pending will be paid.
- ❖ **Applied Trainer:** On this page, the admin will see all the applications for Trainer. You can show tabular/card.
 - 👉 There will be an action icon (👁️). Clicking the icon will open a **modal** and show the details info about the applied person. There will be a **confirmation** and **reject** button.
 - 👉 **confirmation button:** On clicking the **confirmation** button the applied person's status will be changed from normal user to Trainer and that particular entry will be removed from the **Applied Trainer**.
 - 👉 **reject button:** See the bonus section
- ❖ **Balance:** on this page,
 - 👉 Show the Total Remaining Balance and total payment.

👉 Add a pie chart/bar chart or any other chart where you have to show total newsletter subscribers vs total paid members.

👉 show the 6 payments done by members.[More see Optional Part]

Trainer 🧑🏫🧑🏫 (Private)

A Trainer will see The following routes/pages 📌

❖ **Manage Slots:** on this route trainer will see all slots.

👉 If the slot is booked trainer will see that particular slot info including who booked that slot. [more see bonus part]

- ❖ **Manage member**: In this route trainer will see all of his members in card/tabular format. From here trainer can send instructions via email.
- ❖ **Add new Forum**: both **admin** & **trainer** will get access to this page. Both of them can add a new forum. All added forums by the admin and trainer will be shown on the **Community/Forums** page. **[more see bonus part]**
- ❖ **Add new Class**: On this page, there will be a form where a trainer can add a new class. What you add is totally up to you. Added class will be shown on the **Classes Page**.

member **(Private)**

A member will see The following routes/pages

- ❖ **Activity Log**: on this page, the user will see his/her today's activity. Including his current Trainer, his/her training time (slot), and other info. How you design this page depends on you but try to make it relevant.
- ❖ **Profile Settings**: Allows users to manage their account details. He/she can modify anything except the email address.
- ❖ **Recommended Classes Page**: on this page, you have to show some recommended classes for users

★ **[Optional: Add Two more relevant pages/routes here.]**


Login and registration systems


Registration Page

Create a Registration page will have the Email/Password form having the following fields:

- Name
- Email

- Photo URL
- Password



 Note: Make sure you add the user information in a MongoDB collection after successful registration and use them as per your need.

 *Do not enforce the email verification method, as it will inconvenience the examiner. If you want, you can add email verification after receiving the assignment result.*

Login Page +:

When a user clicks on the login button, they will be redirected to the login page which has the following:

- Email/Password
- A Social Login System
- A link that will help the user toggle the login and **registration page**

 Both Registration and Login pages, display relevant error () messages when necessary. Also, A user by default will get the **(role: member)** when he/she registers or sign in with a popup.

Bonus 🎁

🔗 **Reject button**: clicking the reject button will send an email to that particular user. [Explore: [emailJS](#)].

🔗 **Manage Slots**: if the user gets rejected you have to send a response via email. [Explore: [emailJS](#)]. You have the right to implement this part on your own.

🔗 **Community/Forums page**: For each post you have to show the admin/trainer badge for each of their posts. What you implement it's totally up to you. You can get inspiration from Facebook group **admin** and **moderator** posts.

🔗 **404 page**: Create a **404** page. Add any interesting jpg/gif on the 404 page. Do not add header & footer on this page. Just add a jpg/gif & a Back to Home button. The Back to Home button will redirect the user to the home page.

🔗 **Commits & readme** ⚙️

- 📌 Minimum 20 meaningful git commits on the client-side repository.
- 📌 Minimum 10 meaningful commits on the server-side repository.
- 📌 Create a readme on your client-side and add the following info:
 - ✳️ Your project name & live link.
 - ✳️ 5 main Features of your projects with bullet points

🔗 **Fix your Reload & Error Issue** ↺


📌 If you reload the protected/private routes (after login), it will not redirect the user to the login page.

📌 If you reload the website, it will not show the site not found. (Netlify & surge).

🔗 **Website responsiveness** 🖥️

- 📌 Make the Homepage fully responsive (mobile, tablet, and desktop)
- 📌 Optional: if possible make the whole website responsive.


Website Naming

 Give your website a name. The website title will be changed according to your route/page. Suppose your website name is PHero. Then, on the 'login' router/page, your website title will be 'PHero | Login'.

Private Routes Authorization

Upon login, you will create a JWT token and store it on the client side **localStorage**. You will send the token with the call and verify the user. Implementing 401 and 403 for invalid/unauthorized access. 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 all of your private routes.

Environment Variables

 Use the Environment variable to hide the Firebase config keys and MongoDB credentials.

✨ **Optional (But Highly Recommended)** ✨

✨ **Trainer Slot:** If anyone booked that slot it will show unavailable.

✨ **Admin Balance:** Here you have to show the last 6 payment transactions done by members.

✨ **Weekly schedule in Classes:** you can add a tab system. There will be 7 days. Clicking each day will show the activities of that day.


✨ **Featured classes:** You need to count the number of members joining a class. 6 cards will show on the homepage based on the total number of joined members.

Guidelines

★ It's a very very big project so don't try to solve all of your problems at a time. Break a big functionality into many little pieces then try to solve them one by one.

- ★ Do not waste much time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
- ★ Divide the whole task into 3 parts: easy, medium, and hard. Keep easy tasks between your medium and hard tasks. When you get stuck on difficult tasks, do the easy ones. Then again try to solve difficult, medium tasks.
- ★ Don't do coding for a very long time. Take a 5-10 minute break every 40/50 minutes. Give your brain a break to think of new ideas for your task.
- ★ Be strategic about the electricity issue.
- ★ Just don't copy code from ChatGPT try to understand how it works and how can you implement it in a better way.

What needs to submit

1. **Assignment Category:** assignment12_category_0009
 2. **Admin email:**
 3. **Admin password:**
 4. **Front-end Live Site Link:**
 5. **Client Side Github Repository Link:**
 6. **Server Side Github Repository Link:**
-  **Admin credentials: (email, password)**

 **All The Best** 