

Assignment_ID: assignment12_category_0015

Product Hunt Website

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 fulfill 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
- Bistro Boss
- StayVista
- SoloSphere
- Any conceptual session projects

- Any of your previous assignments

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

Website Themes: (Product Hunt Website)

It is a platform where users can discover and share their **tech products**. Tech Products means Web Apps, AI tools, Softwares, Games, Mobile Apps etc. This platform is similar to the <https://producthunt.com> website. Users can submit new products, upvote or downvote existing products & post reviews for products. The platform includes user authentication, roles for normal users, moderators, and admins, as well as a product moderation system. Additionally, a payment system is integrated, allowing users to unlock premium features or obtain extra facilities through paid subscriptions.

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.

User Roles:

1. Normal Users:

- Can browse and view the latest tech products.
- Can upvote products.
- Can submit new tech products for review.

- Can Report, Review other users products.

2. Moderators:

Moderators have the following responsibilities:

- Review and approve/reject submitted products.
- Handle reported products.
- Can mark products as featured.

3. Admins:

- Can manage user roles.
- Can monitor site activities through a statistics page in the dashboard.

Main Requirements

Homepage:

1. Navbar:

Navbar will contain logo/website name & the following navigation links:

- Home
- Products
- Login/Registration button. (Only visible if the user is not logged in)
- User profile picture. (Only visible if the user is logged in).
- If the user clicks on the profile picture, a drop-down will appear with the following items: User name (not clickable), Dashboard, and Logout button.

2. Banner/Slider/Carousel

- You must use Appropriate, Beautiful & Meaningful background images, images and meaningful texts to decorate this section.
- Whether you use a banner or slider it's up to you.

3. Featured Products Section:

Display featured product cards in this section. You must fetch these products from the database. You must keep these facts checked while developing this section:

- At Least 4 product card (Featured Products only)
- Sort cards based on timestamp. (Latest products first)
- Each Card should have the following fields:
 - Product Image
 - Product Name
 - Tags
 - Upvote Button
 - Show total vote value as button text along with a meaningful vote icon.
 - This button will be disabled for the product owner.
 - Only logged in users will be able to interact with this button.
- Card layout and design is up to you.

After clicking on the Upvote button, if the user is not logged in, he/she will be redirected to the login page. If he/she is logged in then the vote value will be increased by 1. A user can vote on a product once only. After clicking on the Product Name, users will be redirected to the [PRODUCT DETAILS](#) page.

4. Trending Products Section:

Display 6 product cards in this section. You must fetch these products from the database. You must keep these facts checked while developing this section:

- Sort cards based on vote count.
- Each Card should have the same info fields and functionality as [Featured Product](#) cards mentioned above.
- Card layout and design is up to you

Below the cards, there should be a button titled **Show All Products**. Clicking on this button will redirect the user to the [PRODUCTS](#) page.

5. Footer:

Create a reasonable and meaningful footer. (including website logo+name, copyright, some contact information, social media links, address, etc.)

Authentication:

On the Registration and Login pages, display relevant error messages when necessary.

1. Login Page:

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

- Email/Password
- Google Sign-in
- A link that will redirect to the registration page

2. Registration Page:

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

- Name
- Email
- Password
- PhotoURL

Note: Do not enforce the email verification method and forget & reset password method, as it will create inconvenience to the examiner. If you want, you can add after receiving the assignment result

Product Details Page (Private Route):

This page (/product/:id) contains the following sections:

- Product Details Section
 - Show Product details such as name, image, description, tags, external links, upvote count etc.
 - Upvote button
 - Button functionality is same as [Featured Products](#) Section card's upvote button.
 - Report Button
 - Users can report a product by clicking on this button.
- Reviews Section

- Show all the reviews of this product as a card/slider in this section.
- Post Review Section
 - A user can post a review of this product. Create a form for collecting review data. Form contains the following fields:
 - Reviewer Name (Read-Only Input field. Get value from context.)
 - Reviewer Image (Read-Only Input field. Get value from context.)
 - Review Description
 - Rating
 - Submit Button

After clicking the **Submit** button, save product id(_id) as productId with all the information mentioned above in the review collection(create a new collection in db).

Products Page:

All the accepted products will be visible on this page.

- Users can search products on this page through the help of a search bar on top of this page.
- Your search word should be based on the tags you used in your product. The search implementation functionality should be done in the backend.
- Below the search bar display 3 cards in each row.
- Implement Pagination in this page. Show 6 cards only on each page.
- Each Card should have the same info fields and functionality as [Featured Product](#) cards mentioned above.

User Dashboard (Private Route):

Note: This must be a dashboard layout.

When a user clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- My Profile
- Add Product
- My Products

1. My Profile Page:

This page will have the following informations:

- User's Name
- User's Image
- User's Email
- Membership Subscribe Button
 - Only visible if he/she has not subscribed yet.
 - Show subscription amount as button Text.
- User Membership Subscription Status (Visible only if he/she is subscribed)
 - Status: Verified

After clicking on the **Membership Subscribe** button, you can redirect the user to a payment page or open a modal in this page containing the payment checkout form. Once he/she successfully pays the amount, redirect them back to this page or close the modal if you use a modal.

2. Add Product Page:

A user can add a product through a form. The form will have the following fields:

- Product Name field (Required).
- Product Image Input field(Required).
- Description (Required).
- Product Owner Info. These fields will be visible but disabled. Take value from context. (Required)
 - Owner Name
 - Owner Image
 - Owner Email
- Tags input field (Use react-tag-input npm package)
- External Links (Website link or landing page link of the product)
- Submit Button

After clicking the Submit button, product data will be saved in the mongodb database. Then show a toast and redirect the user to [My Products](#) page. When you are saving a product in the database, save the timestamp(current time) using the new Data() method along with the data, later you will use this timestamp to sort products in the Featured Products Section.

3. My Products Page:

If a user visits this page, he/she will be able to see all the posts he/she posted. Show them in tabular form. Each row will have:

- Product Name
- Number of votes
- Status
 - Initially status will be pending. Once a moderator reviews the product and sets the status to either Accepted/Rejected, the corresponding status will be visible here.
- Update Button
 - After clicking on this button, the user will be redirected to the update page and there he/she can update product info.
- Delete Button
 - After clicking this button, the product will be removed from both the database and this page.

Moderator Dashboard (Private Route):

Note: This must be a dashboard layout

When a moderator clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- Product Review Queue
- Reported Contents

1. Product Review Queue Page:

This page will have all the posted products data in tabular format. Sort the table data based on product status. Pending products will come first on the table. Each row of the table will have the following information:

- Product Name
- View Details Button
 - Clicking on this button will redirect the moderator to the [Product Details](#) page.
- Make Featured Button

- Clicking on this button marks the product as featured and the product will be visible in the [Featured Products](#) Section on Homepage.
- Accept Button
 - Clicking on this button will change the product status to Accepted.
 - Disable the Accept button afterwards. You can keep the Reject button enabled or disabled.
- Reject Button
 - Clicking on this button will change the product status to Rejected.
 - Disable the Reject button afterwards. You can keep the Accept button enabled or disabled.

2. Reported Contents Page:

This page will have all the reported products data in tabular format. Moderators can review and delete reported products on this page. Each row of the table will have the following information:

- Product Name
- View Details Button
 - Clicking on this button will redirect the moderator to the [Product Details](#) page.
- Delete Button
 - After clicking this button, the product will be removed from both the database and this page.

Admin Dashboard (Private Route):

Note: This must be a dashboard layout

When an Adminr clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- Statistics Page. (See Bonus Section)
- Manage Users
- Manage Coupons. ((See Bonus Section)

1. Manage Users:

Show all the users in a tabular form where each row will have:

- User name
- User email
- Make Moderator
- Make Admin

The admin can make a user Admin or Moderator by clicking on the Make Admin or Make Moderator button.

Other Requirements

1. Add a 404 Error Page.
2. Implement JWT on login (Email/Password and social) and store the token in **Local Storage**.

Challenges Requirements

1. Admin Statistics Page:
 - Create a pie chart to show the total number of products, numbers of reviews, and number of users of the entire site. Don't use other charts instead of a pie chart.
2. Manage Coupon Page:
 - In this page admin can add, view, edit & delete coupons. All operations will be performed in the database through the server, no fake data.
 - You can show added coupons in tabular or card format.
 - Each coupon card will have a coupon code, expiry date, coupon description & discount amount.
 - Coupon code Add Form will have the following fields:
 - Coupon Code
 - Expiry Date
 - Coupon code description
 - Discount Amount
3. In the Homepage, under the Trending Products section, add a slider/carousel for advertising valid coupons that are added by admin. In each slide display coupon

code, discount amount, expiry date & coupon code description. Explore ecommerce websites for motivation, idea of this section.

4. Users can use coupon codes to get discounts while purchasing Membership subscriptions.
5. A normal user can add only 1 product. If he/she purchase the Membership Subscription then the product adding restriction/limit will be removed.

Optional Task

Implement any tasks from the following optional list:

1. Implement a down vote button along with the existing up vote button in each product card.
 - Make the button functional.
2. In the products page, above the search bar, implement a dynamic slider. Here advertise and highlight popular products (products with 10+ votes) in each slide. You can use the term “Rising Products” in this section.
3. 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.
4. Explore and implement any of the animations from the Framer Motion
5. Explore and implement Tanstack query mutations in your api.
6. Add one extra feature of your own. This will help you in the future to differentiate your project from others.

What to Submit

- Assignment Category/variant:
- Admin email:

- Admin password:
- Front-end Live Site Link:
- Client Side Github Repository Link:
- Server Side Github Repository Link: