Pet adoption Platform

assignment12_category_0008

Welcome to the exciting world of creating a Pet Adoption Website using the MERN stack! In this project, your task is to build a user-friendly platform connecting potential pet adopters with adorable companions in need of a forever home.

Main Requirements

Homepage

1. Navbar
    1. Navbar will have the logo of the website, make sure the logo looks good and meaningful
    2. Relevant links to navigate to the website like Home, Pet Listing, Donation Campaigns, Login/Register
    3. Currently logged in user’s profile picture and a drop down menu should appear on clicking the Profile picture, that dropdown menu will have a link to the dashboard and a logout button.
2. A banner section, it can be a normal banner or slider or something else if that suits the website’s theme
3. Pets category section, this section will have button/cards/links to category of pets, such as Cats, Docs, Rabbit, Fish and more if required.
4. Call to action section, this section encourages people to adopt pets and give them a better life. This section will have inspirational images and text
5. About us section, this section will have a short introduction about how the website works and why this website was made
6. Add 2 more sections that is relevant to the theme of the website

Pet listing

1. This page will show all the pets that are not adopted. You can show the pets in a 3 column grid layout with cards
2. All the cards will be shown sorted by date in descending order
3. Add a search field on top of the cards where users can type and search for pets using name, and a dropdown for select menu for seeing pets of a specific category
4. In the cards show the following information of the pet
    1. Pet image
    2. Pet name
    3. Pet age
    4. Pet location
    5. A button for viewing details of the pet
5. implement infinite scrolling in this page, that means as you scroll the page more pets will appear at the bottom

Pet Details

1. Show all the details of the pet and an `Adopt` button.
2. Clicking adopt button will open a modal with a form, the form will have input fields for the user to input some details about
    1. User name, this field's value will be automatically filled and disabled so users can't edit it
    2. Email, this field's value will be automatically filled and disabled so users can't edit it
    3. Phone number, user can write his phone number so the person who is giving adoption can see it
    4. Address, the address of the user who
    5. Submit button
3. After submitting the form a adoption request will be saved into the database with the information provided in the form

Donation campaigns page

1. This page will show all the donation campaigns in a 3 column grid layout.
2. The cards will have Pet name, pet image, maximum donation amount, donated amount and view details button
3. All the cards will be shown sorted by date in descending order
4. implement infinite scrolling in the page, this means as you scroll the page more donation campaigns will appear at the bottom

Donation details

1. In this page all the details of the donation will be shown with a donate now button
2. Clicking the donate now button will open a modal with a input field for Donation Amount and a Credit Card element provided by stipe
3. User can enter the amount of donation he want to make and their card details
4. On submitting the donation a donation will be made and the person who asked for donation will be able to see who donated and how much donated on their Donation Campaigns page
5. After the donation details show a recommended donation section where 3 more active donation campaign will be shown

Authentication

1. Implement email and password authentication and show relevant error while logging in with email and password or register
2. In the registration page, add a input field for the image of the user and full name of the user
3. use the updateProfile function from firebase to update image and name of the user
4. implement 2 additional login method such as google, facebook, github or anything else
5. show relevant errors when logging in with the other methods
6. after successfully registering, save the information of the user such as name and email in the database and store additional boolean property named `role`. Default role will be user and admin can make any other user admin from admin dashboard
7. Make sure to implement jwt on login and store the access_token of the user in either localStorage or cookies.
8. There should be 2 roles for the users. `admin` and `user`

User Dashboard

Make the dashboard with a sidebar and top navbar layout

1. User dashboard will have the following pages
    1. Add a pet
    2. My added pets
    3. Adoption Request
    4. Create Donation Campaign
    5. My Donation Campaigns
    6. My Donations

Add a pet (protected)

1. Logged in users can add their pet using a form and the form will contain the following input fields
    1. Pet image, this will not be a text field where people can put image links, this must be a file select field, use Cloudinary api to automatically upload image when user selects a file
    2. Pet name
    3. Pet age
    4. Pet Category, this will be a dropdown menu containing all possible category of pets, do not use just <select>, rather use a component library such as react-select
    5. Pet location, from where the pet can be picked up when someone wants to adopt it,
    6. Short description, a short description about the pet or small note from the pet owner
    7. Long description, this will be a textarea input field and detailed information about the pet can be written here.
        1. Pro Tip [optional]: You can use WYSIWYG text editors or Markdown text editors (search on google) for the long description
    8. Submit button, clicking this button will submit the form and add the pet into the database.
2. Make sure to use `formik` for handling form submit
3. Handle any kind of errors during form submission and show the error directly under the input field
4. When adding the pet into the database make sure to store the date and time when the pet was added
5. Also add an additional property named `adopted` into the database and make it false by default. This will become true when someone adopts the pet

My added pets (protected)

1. In this page logged in users can see all the pets they added
2. Show the pets in a Tabular format
3. Use Tanstack Table for showing data and make all the columns of table sortable by clicking the column names
4. The table will show Serial Number, Pet name, Pet category, Pet image, Adoption Status, and three buttons
    1. Update Button, clicking this button will redirect the user to the pet update page.
    2. Delete Button, clicking this button will show a modal, with a relevant message and Yes and No button, clicking yes will delete the pet.
    3. Adopted button, Clicking this button will mark the pet as adopted and make the value of the adopted field in the database to `true`
5. Adoption status will show `Adopted` if the pet is already adopted and `Not Adopted` if it is not adopted

6. Make sure to add pagination to the table and show maximum 10 pets at a time, if there are more than 10 pets then the pagination will appear, otherwise it will be hidden

Update pet (protected)

This page will be exactly the same as the `add a pet page`. The only difference is when editing the pet all the fields will be previously filled with the data of the pet that's being edited

Create Donation Campaign (protected)

In this page there will be a form with the following input fields

1. Pet picture, the picture of the pet, must be a file input field (use Cloudinary api for uploading image)
2. Maximum donation amount, highest total amount of donation users can make
3. Last date of donation, the last date when the donation will be closed and people can no longer make donations
4. Short Description
5. Long description
6. Submit button, clicking this button will create a donation campaign by adding all the information to the database
7. When adding the donation campaign into the database make sure to store the date and time when the donation campaign was created

My donation campaigns (protected)

1. In this page logged in users can see the donations they have asked for
2. The data will be shown in a tabular format containing the following columns ⇒ Pet name, Maximum donation amount, donation progress bar that shows how close it is to fulfill the donation requirement
3. Users will not be able to delete any donation from this table. They can only edit the donation info.
4. Pause button, this will pause the donation and no one can donate on that donation requirement until its unpaused.
5. Edit button, clicking this button will redirect the user to edit donation page
6. View Donators button, clicking this button will open a modal that shows the list of user who have donated with the donation amount they donated

Edit donation (protected)

This page will be exactly the same as the `Ask for donation` page. The only difference is when editing the donation all the fields will be previously filled with the data of the donation thats being edited

My Donations (protected)

This page will show a table where logged in users can see where they have donated. The table will have the following columns.

- Pet image
- Pet name
- Donated amount
- Ask for refund, clicking this button will remove your donation from the donation campaign you have donated for

Adoption request (protected)

1. Show a list of all the adoption request of the pets that the currently logged in user added, in a tabular format
2. User can accept or reject the adoption request and he can see the name, email, phone number and location of the person who requested to adopt the pet

Admin Dashboard

If the logged in user is an admin he can see all the pages a user can but also see the additional following pages. if a `user` tries to access admin dashboard pages he will be redirected to user dashboard automatically

Users (protected)

1. show a list of all the users in a tabular format of the users who logged in/ registered into the website
2. in the table show name, email and profile picture of the user, and a `Make Admin` button. Clicking the make admin button the user's role on dashboard will become `admin` and the user will be able to access admin routes too.
3. [Optional] if you can you can add a ban button to ban a user so that user will not be able to log in to the website and if he tries to login he will get a message that his account was banned

All Pets (protected)

1. Show a list of all the added pets by all users in a tabular format
2. admin will be able to delete or update any pet added by any user
3. admin will be able to change the pet's status to adopted or not adopted if necessary

All Donations (protected)

1. Show a list of all the donation campaigns added by all users
2. admin will be able to delete, edit, and pause any donation campaign added by any user
3. a paused donation campaign will show on the website but no one will be able to donate until unpaused again

Additional Information (Not a page)

1. Admins also can access the user routes on dashboard

Optional Task

1. Do no use daisy ui for designing your website, try to make your own website using just TailwindCSS if you can
2. if you are using a component library then make sure to customize the components rather than just copy and pasting the components without any customization
3. Make sure the website looks good in all screens (mobile/tablet/laptop/desktop)
4. The color combination of the website should be simple, not too colorful
5. For all the select menu in the website use react-select or similar component library
6. For the tables try to use tanstack-table or any other data-table library
7. If data does not come from API the website should not crash or show 404 page, it should show a message that data failed to load

Prohibited rules

Do not copy design from previous projects or assignments, such as cards, tables, navbar, forms, etc, doing so will result in a un-considerable penalty, so be carefull

What to submit

1. **Assignment Category:** assignment12_category_0008

2. **Admin email:**
3. **Admin password:**
4. **Front-end Live Site Link:**
5. **Client Side Github Repository Link:**
6. **Server Side Github Repository Link:**