**JOB TASK**

**Task statement:**

**"House Hunter"** is a delightful platform that simplifies the house rental process. It serves as a connecting bridge between house owners, who wish to rent out their properties, and house renters, who are seeking their dream homes. With HouseHunter, owners can eﬀortlessly list their houses for rent, providing detailed information about their properties. On the other hand, renters can easily search for available houses and make bookings tailored to their needs. By leveraging the power of Node.js, React.js, and MongoDB, we aim to build the following features for this exceptional platform:

1. **User Registration:**

* Create a registration form where users provide their full name, role (House Owner or House Renter) (it must be selected as an option), phone number, email, and password.
* Store all the registered user information in the database and make sure not to duplicate any users.
* Implement custom authentication. You need to use JWT and mongodb databases to handle user registration and login.

**[N.B. You cannot use any authentication service providers like firebase etc].**

1. **House Owner Dashboard:**

* Upon successful registration as a House Owner, users should be redirected to their dashboard.
* Implement a dashboard where House Owners can manage their listed houses and bookings.
* Display a list of houses owned by the logged-in House Owner.

* Add an "Add New House" button that opens a form/modal where the House Owner can provide all the required information about the house (name, address, city, bedrooms, bathrooms, room size, picture, availability date, rent per month, phone number, and description). All the fields are mandatory, and only Bangladeshi phone numbers are allowed.
* Implement functionality to add new houses to the database.

* Display the list of owned houses in a table with options to delete or edit each house. After clicking on the delete button, it should be removed instantly from the table. After clicking on the edit button, all the fields should be filled automatically so that they can edit the previous information and instantly show the newly updated information in the table.

**3. House Renter Dashboard:**

* Upon successful registration as a House Renter, users should be redirected to their unique dashboard.
* Create a dashboard where House Renters can manage their bookings.

* House Renters should be able to book houses from the home page by providing their name, email, and phone number (name and email should be filled automatically and can't be modified).
* Implement functionality to store the booking details in the database and associate them with the House Renter.
* Display the booked houses on the House Renter dashboard, allowing House Renters to view their current bookings.
* House renters can book a maximum of two houses. Implement the option to remove or delete booked houses and allow House Renters to free up space for new bookings.

**4. Home Page and House Search:**

* Create a home page that displays all the listed houses.

* Anyone can search for houses, but only House Renters can book houses for rent and need to log in to book a house.

- Implement functionality to search houses and filter by city, bedrooms, bathrooms, room size, availability, and rent per month (using a range selector). Filter the appropriate houses based on user preferences. You can try to implement filters from the backend, but it's not mandatory.

* Implement server-side pagination/infinite scroll to fetch only 10 data at once. You can’t show more than 10 data at once if you implement pagination. For infinite-scrolling you can use any npm packages.

**Note:**

* The website's color combination should be perfect.

**Extra (optional):**

* Try to implement using Mongoose (ODM).

* Use Vite React for the front-end.

Remember to handle errors, validate user input, and provide appropriate feedback to users throughout the application. This outline provides a simplified overview of the tasks involved in building the HouseHunter project using the specified technologies.

* **You have to deploy your site and provide us with the live link and github link. Don’t forget to**

**make meaningful commits ]**

**Solve the problem in javascript: - 10**

**Valid Palindrome**

Write a JavaScript function called isPalindrome that takes a string as input and returns true if the string is a palindrome otherwise return false. A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward, ignoring punctuation, case, and spacing.

**Sample:**

isPalindrome("level"); // Output: true

isPalindrome("hello"); // Output: false

isPalindrome("A man, a plan, a canal: Panama"); // Output: true

**Some Rules (Read with Care):**

1. Your time limit will be from **January 22, 2024 to 12.30 PM January 24, 2024**. After that, we will not accept any submission.
2. You can't submit multiple times, but if you do then only your first submission will be accepted.

1. You have to submit your solution in this google form.

1. You can't take any help from our instructors, Job placement team, your course support groups & programming hero community. If you are caught taking help from support sessions, Job placements team or asking for a solution in support groups & community your application will be rejected.

1. Every task will be assessed carefully by our web instructor team, if you are caught copying or sharing codes with other candidates you and the other candidates with whom you shared your code both's application will be rejected and you will be banned and blacklisted from all programming hero's future job openings.
2. Remember this is not a course related assignment or practice task, this is a serious job task and we expect you to respect it and complete the task in proper time.