Slide:

Slide 1.2 scope:

+ Thêm Filter listing by category, min max price and type of place.

Slide 4.1: Technology

+ **Cron jobs org** is a service that allows users to easily set up and manage automated tasks (cron jobs). When a user configures a cron job, the service automatically triggers **frontend (FE)** to send a request to the **backend (BE)** of the system. The **backend** then processes the request and executes the scheduled tasks as defined by the user.

This service helps **keep the backend "awake"** and ready to receive and process cron jobs on time, without the user needing to worry about managing the infrastructure or maintaining the backend's state. This could include actions like sending HTTP requests, running scripts, or performing other automated tasks.

In essence, **cron jobs org** facilitates the **frontend** in triggering automated tasks on the **backend**, while ensuring the backend remains operational and ready to handle jobs without interruption.

**Cron jobs org** là một dịch vụ giúp người dùng dễ dàng thiết lập và quản lý các tác vụ tự động (cron jobs). Khi người dùng cấu hình một cron job, dịch vụ sẽ tự động gọi **frontend (FE)** để gửi yêu cầu (request) đến **backend (BE)** của hệ thống. **Backend** sau đó xử lý yêu cầu và thực thi các tác vụ định kỳ mà người dùng đã thiết lập.

Dịch vụ này giúp **giữ backend luôn "tỉnh"** và sẵn sàng nhận và xử lý các cron jobs theo lịch trình đã định mà không cần người dùng phải lo lắng về việc quản lý cơ sở hạ tầng hoặc duy trì trạng thái của hệ thống backend. Điều này có thể bao gồm việc gửi yêu cầu HTTP, chạy các script, hoặc thực hiện các tác vụ tự động khác.

Về cơ bản, **cron jobs org** giúp **frontend** kích hoạt các tác vụ tự động trên **backend**, đồng thời duy trì backend hoạt động ổn định và sẵn sàng xử lý công việc mà không bị gián đoạn.

This diagram illustrates the architecture of a web application deployment process using various technologies. Here’s a breakdown of the components:

1. **GitHub**: This is where the source code (repositories) of the application is stored and managed.
2. **Render**: This platform is responsible for deployment. It takes the code from GitHub and deploys the application to be accessible via the web.
3. **React**: This is a JavaScript library used for building user interfaces, particularly for single-page applications. In this setup, it serves the static site.
4. **Express**: This is a web server framework for Node.js that handles requests and responses. It acts as the backend server for the application.
5. **MongoDB Atlas**: This is a cloud database service for hosting MongoDB databases. It stores and manages the application’s data.
6. **Browser**: This represents the end-user's interface through which they can access the deployed web application.

**Workflow:**

* The application’s code is pushed to GitHub.
* Render takes this code and deploys it.
* Users access the React static site through their browsers.
* The Express server handles any backend logic and communicates with the MongoDB Atlas database for data storage and retrieval.

Demo:

Here is the beautiful fullstack property website. I will show you the demo of the entire application.

+ On the top right here we have our Logo, search bar and user navigation. When there is no user login yet we can log in and sign up. So let’s register

+ This is register page you can register a user. When the password and confirm password does not match. This line will appear.  
+ We can upload the profile photo and then register.

+ It will lead us to the login page. Now we can log into the account we just created.  
+ It will lead us to the homepage and now we have all the list of the user.

+Below we have all the top categories, click into this we have all the properties belong to category we choose( chỗ explore top categories có các options cho chọn và sau khi click vào thì ra 1 vài thành phần bên trong đó)

+ Below we have all categories listing here(1 list categories), we can filter the listings belong to the categories we choose.

+ If you interested in this listing you can click the heart to add this listing to the wish list. The heart will become from white to pink show that it already added it to the wishlist. You can check in your user navigation – The wish list.

+ Come to the homepage, you can see all of the photos come and back. You can also click anywhere around the listing -> listing details(can see the titles, the images, the place, the number of guests, the bedroom, bath allow in this property, who hosted the property, description, highlight and utilities the place offers. Right here we have the calendar that you can pick the start date and end date. It can calculate how many nights and the total price if you want to book this place click booking.

+ After booking, Lead to your trip list, where all the booking you booked. It show you the start date the end date and the total price. If you want to check this just click it it will show back to the listing details.

+ Also you can add this list to the wish list by clicking this heart as well save, the listing also appear in the wish list.

+ If you turn the pink heart to the white heart it will remove from your wish list.

+ Your property list: You need to publish a new place, you click become a host here. In this creating listing page we can choose the category.  
Đến đoạn add hình có thể add nhiều hình và delete từng cái hoặc là move photo di chuyển khắp nơi to the new order và có thể upload more if you want.

Set your price: can apply the price or adjust as well. Check the information is right and click CREATE YOUR LISTING.

+ It will lead you to the property list is all the property you owned. You can click into to check your ideas

+ RESERVATION LIST is all the list that someone else book your property

+ Navbar: You can search anything you want for example you can search a place have a lake click search here. It will show all the properties near the lake, or related to the lake.

Search mountain, filter by categories, mix min price,…