Create a project plan for a full-stack Kanban board application called "KanbanFlow” (the repo is in https://github.com/dnsjm/kanban-flow.git). The tech stack should be: a **React.js** frontend (using Vite, Axios, and Tailwind CSS), a **Java Spring Boot** backend (using Maven, Spring Web, Spring Data JPA, and Spring Security for JWT authentication), and a **PostgreSQL** database. The plan should cover setting up a GitHub repository, building the backend REST API with models, repositories, services, and controllers, then developing the React frontend with components for boards and tasks, including drag-and-drop functionality. Finally, outline the deployment process: hosting the PostgreSQL database on ElephantSQL, deploying the Spring Boot API to **Render**, and deploying the React frontend to **Vercel**, ensuring they are connected via environment variables.

Step 1: Deploy the PostgreSQL Database (ElephantSQL)

We'll start by creating the database that your backend will use.

1. Create an ElephantSQL Account:

\* Go to https://www.elephantsql.com/ (https://www.elephantsql.com/) and sign up for a free account.

2. Create a New Instance:

\* Once logged in, click the "Create New Instance" button.

\* Give your instance a name (e.g., "kanban-flow-db").

\* Select the "Tiny Turtle (Free)" plan.

\* Choose a region that is geographically close to you.

\* Click "Review" and then "Create instance".

3. Get Connection Details:

\* After the instance is created, click on its name in your dashboard to see the details.

\* You will see a connection URL (it looks like postgres://username:password@hostname/databasename).

\* You will need the components of this URL for the next step. Let's break it down:

\* User: The part after postgres:// and before the first :.

\* Password: The part after the first : and before the @.

\* Host: The part after the @.

\* Database: The part after the final /.

Important: Keep this browser tab open. You will need to copy and paste these credentials into Render in the next step.

---

Step 2: Deploy the Backend API (Render)

Now we'll get your Spring Boot application running online.

1. Create a Render Account:

\* Go to https://render.com/ (https://render.com/) and sign up. You can sign up with your GitHub account to make connecting

your repository easier.

2. Create a New Web Service:

\* On the Render dashboard, click "New +" and select "Web Service".

\* Connect your GitHub account if you haven't already, and select your kanban-flow repository.

\* Render will ask for some details:

\* Name: Give your service a unique name (e.g., "kanban-flow-api").

\* Root Directory: backend (This is crucial! It tells Render to look inside your /backend folder).

\* Environment: Java.

\* Region: Choose a region close to your ElephantSQL region.

\* Branch: main (or whichever branch has your latest code).

\* Build Command: ./mvnw clean package (Render might default to something similar, but package is often more reliable than

install for this purpose).

\* Start Command: java -jar target/kanbanflow-0.0.1-SNAPSHOT.jar.

\* Instance Type: Free.

3. Add Environment Variables:

\* Before creating the service, scroll down to the "Environment" section.

\* Click "Add Environment Variable" for each of the following:

\* Key: JDBC\_DATABASE\_URL

\* Value: Go back to your ElephantSQL tab. Copy the full URL and paste it here.

\* Key: JDBC\_DATABASE\_USERNAME

\* Value: Copy the User from ElephantSQL.

\* Key: JDBC\_DATABASE\_PASSWORD

\* Value: Copy the Password from ElephantSQL.

\* Key: JWT\_SECRET

\* Value: Create a long, random, and secret string of characters. You can use a password generator for this. This is

critical for security.

4. Deploy:

\* Click the "Create Web Service" button at the bottom.

\* Render will now start building and deploying your backend. This will take a few minutes. You can watch the progress in the

logs.

\* Once it's live, Render will provide you with a public URL for your API (e.g., https://kanban-flow-api.onrender.com).

Important: Copy this API URL. You'll need it for the final step.

---

Step 3: Deploy the Frontend (Vercel)

Finally, let's get your React app online and connect it to the backend.

1. Create a Vercel Account:

\* Go to https://vercel.com/ (https://vercel.com/) and sign up, preferably with your GitHub account.

2. Create a New Project:

\* From your Vercel dashboard, click "Add New..." and select "Project".

\* Select your kanban-flow GitHub repository.

\* Vercel will automatically detect that it's a Vite project.

3. Configure the Project:

\* Root Directory: Change this to frontend. This tells Vercel where your React code is.

\* Expand the "Environment Variables" section.

\* Add the following variable:

\* Key: VITE\_API\_URL

\* Value: Paste the URL of your Render backend API that you copied in the previous step, and make sure to append /api to

the end of it. For example: https://kanban-flow-api.onrender.com/api.

4. Deploy:

\* Click the "Deploy" button.

\* Vercel will build and deploy your frontend. This is usually very fast.

\* Once finished, Vercel will give you your public URL (e.g., https://kanban-flow.vercel.app).

---

You're Done!

Your application should now be live.

\* Go to your Vercel URL to see the frontend.

\* Try registering a new user and logging in.

\* Create a board, and then click on it to go to the board page.

\* Add some tasks and drag them between columns.

If you encounter any issues, check the "Logs" tab in both your Render and Vercel project dashboards. They are the best place to

find and debug errors.