

Kanban Board Application

Implement the Kanban Board Application. The backend is given, and the UI mockup is shown below.

The image shows a web browser window titled "A Web Page" with a URL bar containing "http://". The main content area is titled "Kanban Board". On the left, there is a sidebar with three categories: "Done", "In Progress", and "Todo". Under "Done", there is a card "Interview Customers" with a count of 2, and a card "Contextual Inquiry" with a count of 2. Under "In Progress", there are three cards: "Design the UI" (2), "Graphic Design" (2), and "Design the Backend" (2). Under "Todo", there are three cards: "Prototyping", "Development" (2), and "QA" (3). On the right, there is a form for creating or editing a card. It has a "Name" field with the value "Interview Customers", a "Status" dropdown menu with the value "Done", and a "Description" text area with the value "Interview at least two customers". Below the form, there is a "Task List" section with a "+" button and two items: "Customer 1" and "Bank 2", each with a checkbox and a delete button (X).

In the package, you will find `workspace.zip`, which is a simple RESTful API implementation using Node.js, Express.js, and MongoDB.

First install Node.js and MongoDB. Then, unzip `workspace.zip` and use `npm install` to install the backend application. (`node server.js`)

`kanban.routes.js` under `app/routes` indicates the following available RESTful APIs (You can use Postman to manually test them.)

- GET `/cards`: lists all the kanban cards
- POST `/cards`: create a new kanban card

- With raw JSON {"title": "...", "description": "...", "status": "..."}
 - GET /cards/:card_id: get information of a specific kanban card, includes its task list
 - PUT /cards/:card_id: update information of a specific kanban card
 - With raw JSON {"title": "...", "description": "...", "status": "..."}
 - DELETE /cards/:card_id: delete a specific kanban card
- POST /cards/:card_id/tasks: create a new task for a specific kanban card
 - With raw JSON {"taskname": "..."}
- PUT /cards/:card_id/tasks/:task_id: update the status of a task in a specific kanban card
 - With raw JSON {"taskstatus": true}
- DELETE /cards/:card_id/tasks/:task_id: delete a task in a specific kanban card

The package also includes a sample data set in the JSON format, `kanban.json`.

Please implement the ReactJS front end so that a user can

- List all kanban cards grouped by its status (`todo`, `in-progress`, or `done`)
- Create a new kanban card by providing the name. The default status is `todo`.
- Update an existing kanban card.
- Delete an existing kanban card.

All the changes in the detail view are saved automatically when

- the Name field is out of focus;
- the selected item in the Status drop down changes;
- the Description text area is out of focus;
- the checkbox of a task entry is clicked;
- a new task entry is added and the input field is out of focus; or
- a task entry is deleted.

Once the changes (the name and the number of tasks) in the detail view are saved, the queue view on the left-hand side will be automatically updated.