Fetch Patient Details:

Here’s a detailed explanation of the \*\*FindByPatientId\*\* component:

---

### \*\*Purpose\*\*

The `FindByPatientId` component is designed to fetch and display the details of a patient based on their `patientId`. The `patientId` is accessed from the shared context (`UserContext`) and passed to an API endpoint to retrieve the patient's data.

---

### \*\*Code Explanation\*\*

#### \*\*Imports\*\*

1. \*\*React, Component\*\*: Used for creating a class-based React component.

2. \*\*Bootstrap Components\*\*:

- `Container`: Wraps content with consistent margins and alignment.

- `Card`: Displays patient details in a styled container.

- `Alert`: Shows a message when no results are found.

3. \*\*UserContext\*\*: Accesses shared context to retrieve the `patientId`.

---

#### \*\*State Variables\*\*

1. \*\*`patientDetails` (default: `null`)\*\*:

- Stores the patient's details fetched from the API.

- If `null`, no data is displayed.

2. \*\*`errorMessage` (default: `""`)\*\*:

- Intended for handling and displaying error messages, though not currently used in the rendering.

3. \*\*`noResults` (default: `false`)\*\*:

- Tracks whether the API call returned no results.

- Controls the display of the "No Patient Details Available" alert.

---

#### \*\*Context\*\*

- \*\*`UserContext`\*\*:

- Defined in `../Login/login.jsx`.

- Provides shared data for components within the app.

- In this case, the `contextType` accesses the patient's ID (`patientId`).

---

#### \*\*Lifecycle Method\*\*

1. \*\*`componentDidMount()`\*\*:

- Invoked once the component is mounted.

- Calls `fetchPatientDetails` to fetch patient details immediately after the component loads.

---

#### \*\*Methods\*\*

1. \*\*`fetchPatientDetails()`\*\*:

- Fetches patient data from the backend API using the `patientId` from the context.

- \*\*Steps\*\*:

- Access the `patientId` from the context using `this.context?.patientId`.

- Call the API at the endpoint: `http://localhost:8080/patient/patientId/{patientId}`.

- \*\*Response Handling\*\*:

- If the response is successful (`response.ok`), parse the JSON data and update the state with:

- `patientDetails`: Patient's information.

- `noResults`: Set to `false`.

- If the response fails, set:

- `patientDetails`: `null`.

- `noResults`: `true` to display the "No Patient Details Available" alert.

- \*\*Error Handling\*\*:

- Logs the error to the console for debugging.

- Updates the state to indicate no results (`noResults: true`).

---

#### \*\*Render Method\*\*

The `render()` function dynamically displays content based on the component's state:

1. \*\*Title\*\*: Displays a centered title, "Patient Details."

2. \*\*Conditionally Rendered Content\*\*:

- \*\*When `patientDetails` is available\*\*:

- Shows a `Card` containing the patient's details, such as:

- `Patient ID`

- `Name`

- `Date of Birth`

- `Phone`

- `Email`

- `Gender`

- \*\*When no results are found (`noResults: true`)\*\*:

- Displays an `Alert` with the message: "No Patient Details Available."

- \*\*When data is still loading (`patientDetails: null` and `noResults: false`)\*\*:

- Displays nothing (silent loading state).

---

### \*\*Key Features\*\*

1. \*\*Context Usage\*\*:

- Accesses `patientId` from the `UserContext`, making it easy to manage shared data across components without passing props.

2. \*\*API Integration\*\*:

- Fetches data from the backend API dynamically using the `patientId`.

3. \*\*Error Handling\*\*:

- Handles missing or invalid patient IDs gracefully by showing an alert.

4. \*\*Dynamic Rendering\*\*:

- Displays different content (`Card` or `Alert`) based on the state (`patientDetails` and `noResults`).

5. \*\*Styling\*\*:

- Uses Bootstrap components for a responsive and clean UI.

---

### \*\*State and Method Relationships\*\*

| \*\*State/Variable\*\* | \*\*Purpose\*\* |

|-----------------------|----------------------------------------------------------------------------------------------|

| `patientDetails` | Stores the fetched patient details to be displayed in the `Card`. |

| `errorMessage` | (Not used in UI) Designed to store error messages during fetch failures. |

| `noResults` | Indicates whether no patient details were found (used to show the "No Patient Details" alert). |

| \*\*Method\*\* | \*\*Purpose\*\* |

|------------------------|---------------------------------------------------------------------------------------------|

| `fetchPatientDetails()`| Fetches patient data from the API and updates state based on the response. |

---

### \*\*Flow Overview\*\*

1. Component mounts.

2. `componentDidMount` triggers `fetchPatientDetails`.

3. `fetchPatientDetails`:

- Fetches patient data using the `patientId` from the context.

- Updates the state with the fetched data or flags an error.

4. `render()` displays:

- Patient details if available.

- Alert if no results are found.

---

Let me know if you'd like further details or explanations!