



## CODE CHALLENGE

### Introduction and Instructions

Your task is to build a simple, full-stack web application given a basic set of functional and technical requirements. We do a lot of web development in React and Node.js at Tempus, so we're looking to see how you build a full web application using these technologies. You're free to add in any additional libraries or other technologies as you like.

We absolutely encourage you to take full advantage of any resources available online to complete these exercises (seed projects, utility libraries, documentation, etc).

Be sure to include a readme file that walks us through your code and explains any assumptions or choices you made while building your solution. Try to provide as complete a solution as you can, but if you don't have enough time for the entire exercise, **please submit your work anyway**, include a brief description of your solution to the remaining portion as well. Expected duration: 4-6 hours

We look forward to checking out your code, good luck!

- The Tempus Engineering Team

## EXERCISE: Doctors and Patients in React and Node.js

Build a simple, but well-crafted and appealing web application that allows doctors and patients to collect and share healthcare information:

- Patients' personal details, including:
  - Name
  - Age
  - Email Address
  - Mailing Address
  - Phone Number

### Functional Requirements

- Doctors and Patients sign in with username / password credentials
- The system recognizes the user as either a doctor or a patient
- Upon signing in, doctors see a listing of patients in the system.
  - Doctors can also search for patients by name.
- When a doctor selects a patient:
  - The patient's personal details are displayed.
- Upon signing in, patients see an overview of their information in the system:
  - Their personal details are displayed, **able to be edited**.

### Technical Requirements

- We're looking for a complete client and server implementation: a browser-based single page application that communicates via HTTP calls to an API to fetch and store data. Use React to implement your web application and Node.js for your API server. You're welcome to use any additional libraries or technologies you like.
- No need to create a user registration process, feel free to seed accounts.
- All data must be **persisted**.