# UI Documentation

## Problem

A powerful front-end technology is needed which is capable of handling API calls to the backend. The technology must effectively send to and receive data from the API. The UI must interact with the three main parties of the project which are Patient, Hospital and Buyer. The parties must have access to the following operations:

1. Patient
   1. Login/Register
   2. View Medical Report
   3. View Profile
   4. View Requests from Buyers
   5. Accept/Decline requests or Deal with the Buyer
2. Hospital
   1. Login
   2. View Patients Records
   3. Add Medical Report
   4. View Medical Report
   5. Edit Medical Report
   6. Delete Medical Report
3. Buyer
   1. Login/Register
   2. View Medical Transactions
   3. Filter Medical Transactions
   4. Request to purchase Medical Report
   5. View purchased Medical Report

## Solution

Angular has selected for development of MyContext Web App. Angular belongs to the MEAN Stack which is one of the popular stacks for developing dynamic web applications. It provides numerous features and benefits making it a suitable choice for this project rather than other technologies such as PHP, Java EE, jQuery, etc. The benefits of Angular are explained below:

1. Powerful Ecosystem  
   There are large number of packages, plugins, development tools, add-ons and other angular resources. These include toolings, IDEs, UI environment, Angular Universal, etc. so there wont be gaps or dead ends which using the technology. Most importantly, it also facilitates ASP.NET using which the API will be created.
2. Two-Way Data Binding  
   It is a beneficial technique to establish relation between business logic and UI. It is easy and useful for making real time adjustments in the web app. When the model is updated, the view is also updated instantly. Furthermore, fast response time and natural development is possible as the app reacts quickly to changes in UI or browser.
3. Consistent Code and Performance  
   Angular helps to lower cost and improve performance. Angular code is resuable, easy to maintain and easily readable. Large variation in code is not required. There are many heavy lifting modules for grids, tables, routing and data handling. So re-inventing the wheel is not necessary.

## UI Architecture

The navigational structure of the web application is illustrated below:

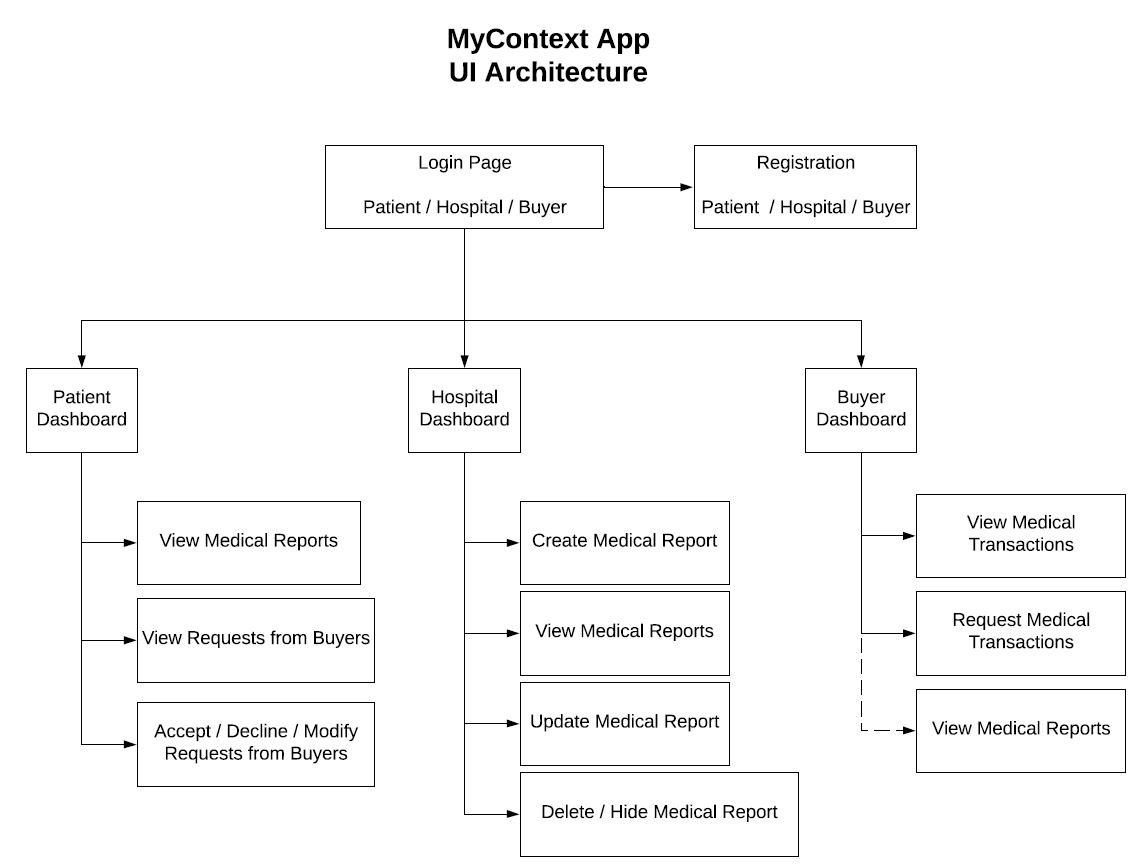


Figure : UI Architecture

## Installation

This guide will explain how an angular project can be started using Angular CLI tool.

### Prerequisites:

* Node.js version 8.x or 10.x. (Download from <https://nodejs.org>)

### Install the Angular CLI

Run the following command in a terminal/console window:



### Create the workspace and initial app

Change directory to where you want the project to be created and run the following command:



### Serve the application

Change directory to inside the project and run the following command to launch the project:

### Bootstrap

Run the following command in terminal/console to install bootstrap:



The CSS installed using bootstrap must be imported. Place this code in src/style.css:



### Toastr Notification

Run the following command in terminal/console to install ngx-toastr:





Place this code in src/style.css:



### Popup Prompt

Run the following command in terminal/console to install np2-opd-popup prompt:



## Project Directory

### Tree Diagram

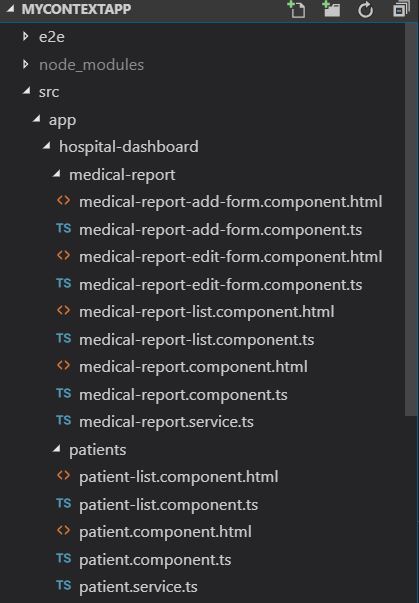
  


Figure : Tree of Project Directory

### File Description

|  |  |  |
| --- | --- | --- |
| Folder | File Name | Purpose |
| src/app/hospital-dashboard/ |  | Grouping of Hospital Dashboard files |
| medical-report/ | * medical-report-add-form.component.ts * medical-report-add-form.component.html | Creates Add Medical Report Page |
| medical-report/ | * medical-report-edit-form.component.ts * medical-report-edit-form.component.html | Creates Edit Medical Report Page |
| medical-report/ | * medical-report-list.component.ts * medical-report-list.component.html * medical-report.component.ts * medical-report.component.html | Creates View Medical Report Page |
| medical-report/ | * medical-report.service.ts | Holds medical report data for CRUD operation pages of medical report |
| patients/ | * patient-list.component.ts * patient-list.component.html * patient.component.ts * patient.component.html | Creates View Patient List Page |
| patients/ | * patient.service.ts | Holds patient data for View Patient List Page |
|  |  |  |
| src/app/patient-dashboard/ |  | Grouping of Patient Dashboard files |
| medical-report/ | * my-medical-report.component.html * my-medical-reports.component | Creates View Medical Report Page |
| medical-report/ | * my-medical-report.service.ts | Holds medical report data for View Medical Report page |
| profile/ | * my-profile.component.css * my-profile.component.html * my-profile.component.ts | Creates View My Profile Page |
| requests/ | * my-request.component.html * my-request.component.ts | Creates View Buyer Request Page |
| requests/ | * my-request.service.ts | Holds request data for View Requests Page |
| assets/images/ |  | Holds profile picture of Patients |

## Dependencies

All Modules, Components and Services used in the Project can be seen in the app.module.ts file below:

# 

Figure : app.module.ts file

## Navigation

After the project is running, these links can be used to access different pages of the web application. The route/link to all the web pages of the application is given below:



Figure : Navigation links

The navigation bar of the web application can also be used to access different pages. Hospital Dashboard and Patient Dashboard has link to its unique pages in its navigation bar.



Figure : Hospital Navigation Bar



Figure : Patient Navigation Bar

## Screenshot of Web App’s UI

### Hospital Dashboard

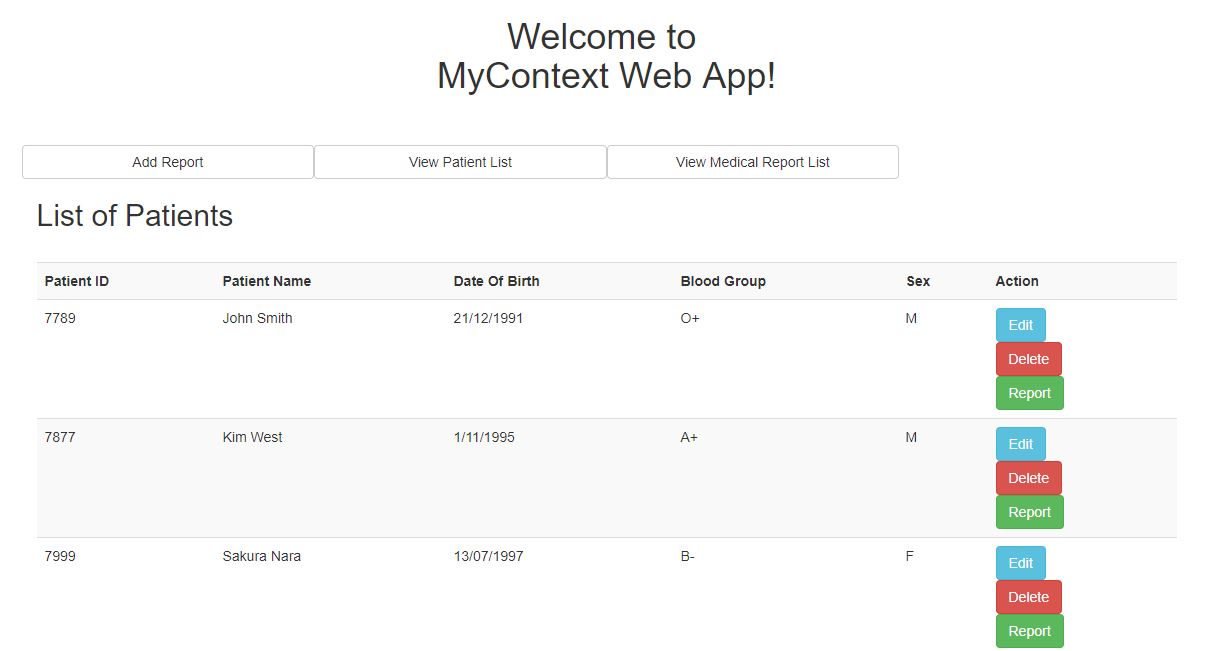


Figure : View Patient Details page

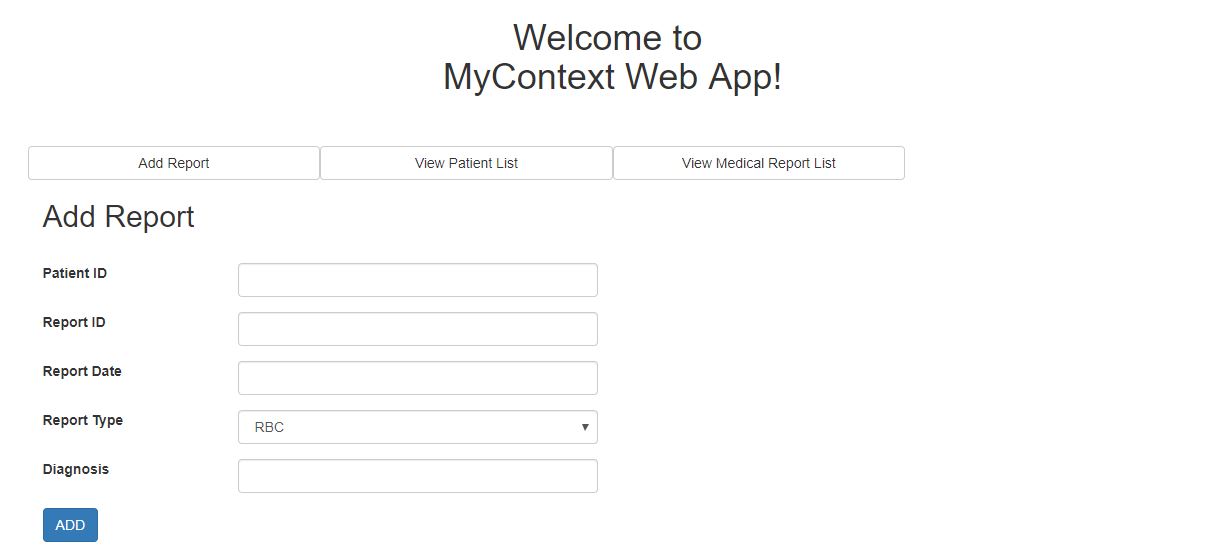


Figure : Add Medical Report Page

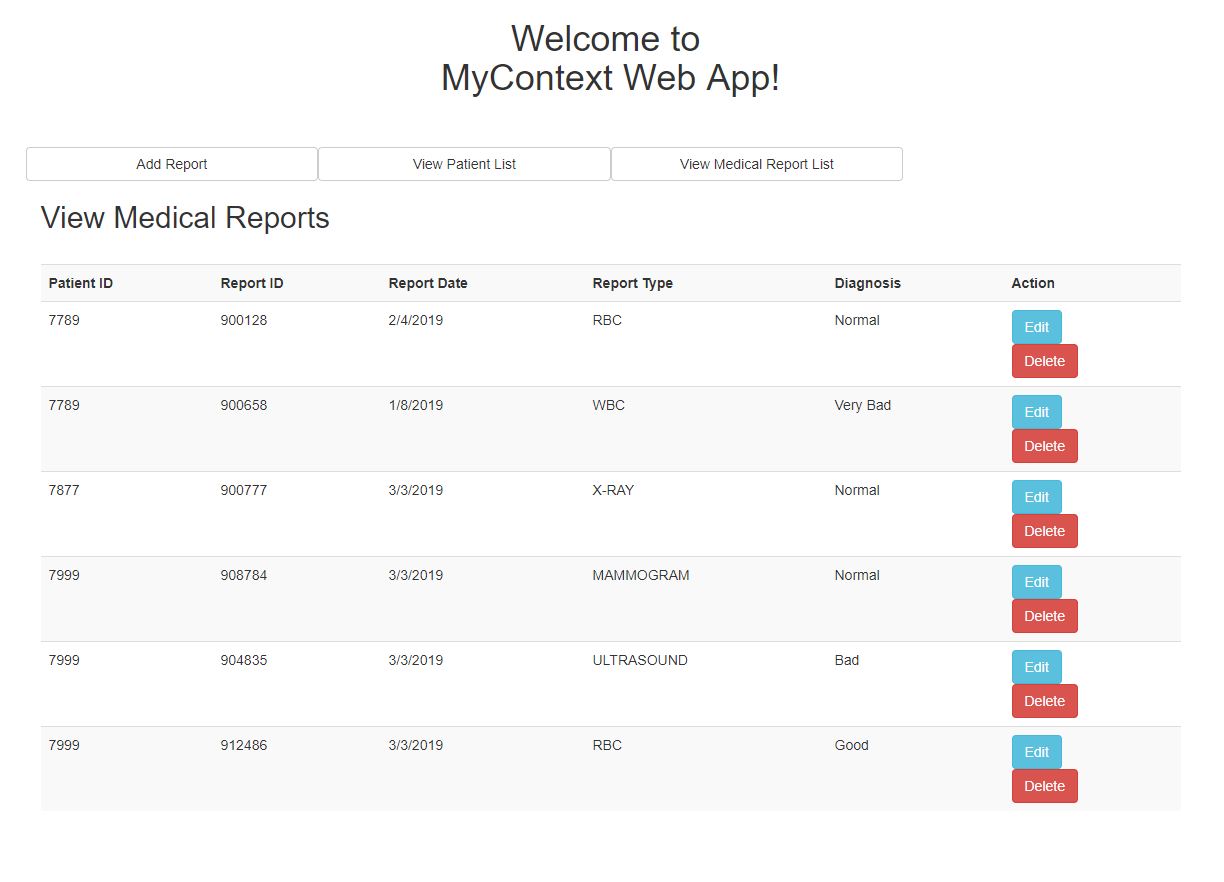


Figure : View Medical Report Page

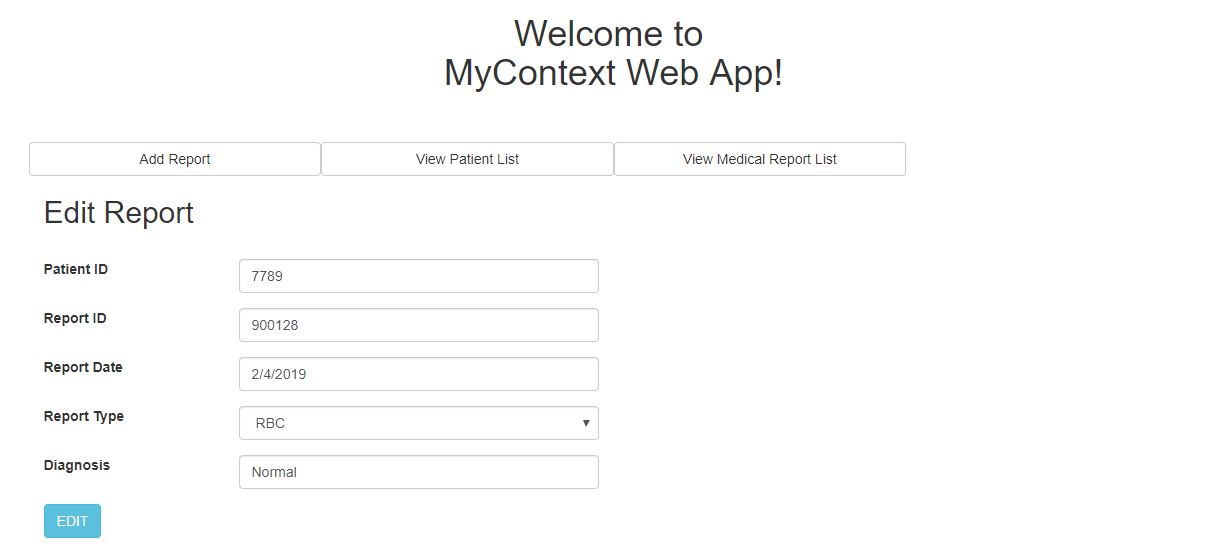


Figure : Edit Medical Report Page

### Patient Dashboard

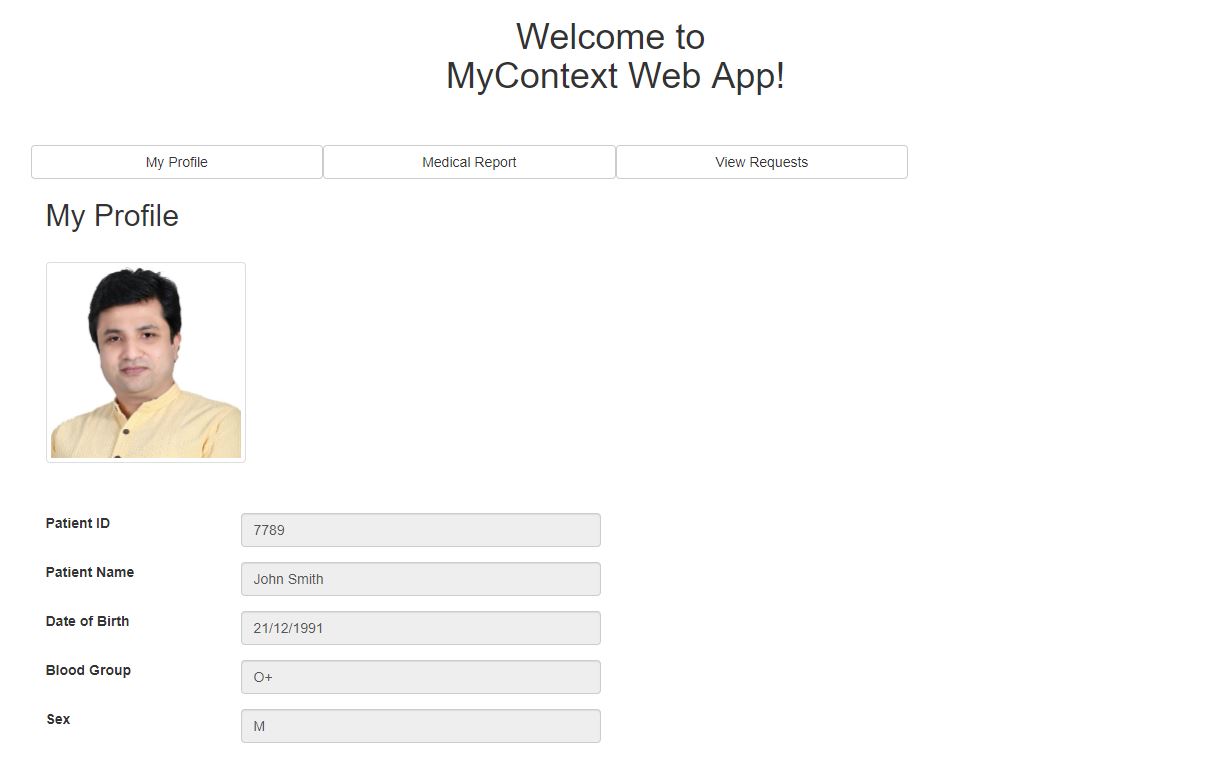


Figure : View Profile Page

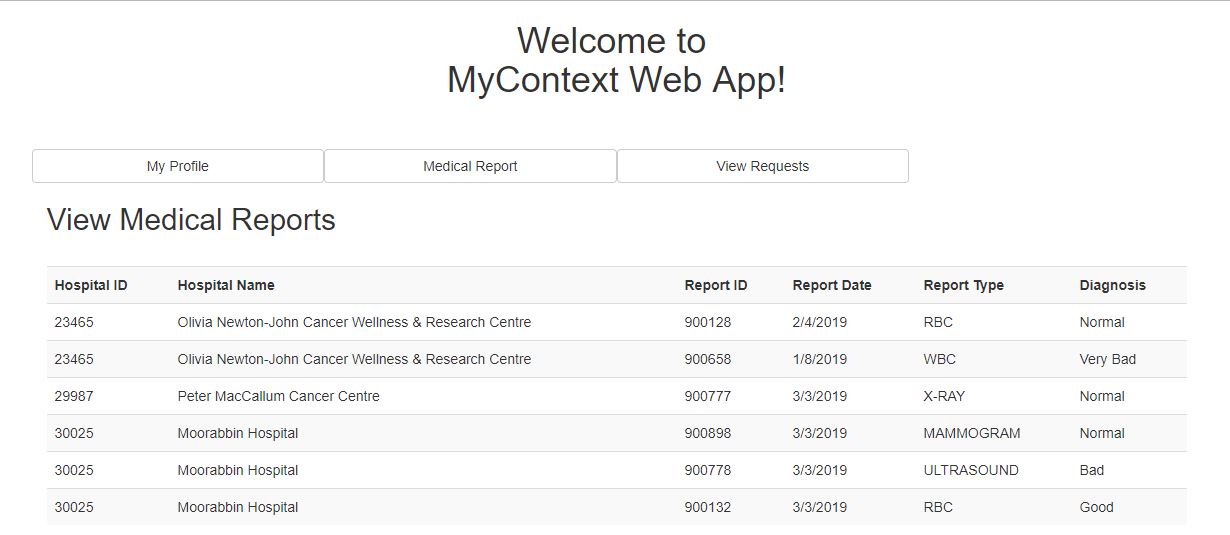


Figure : View Medical Report Page

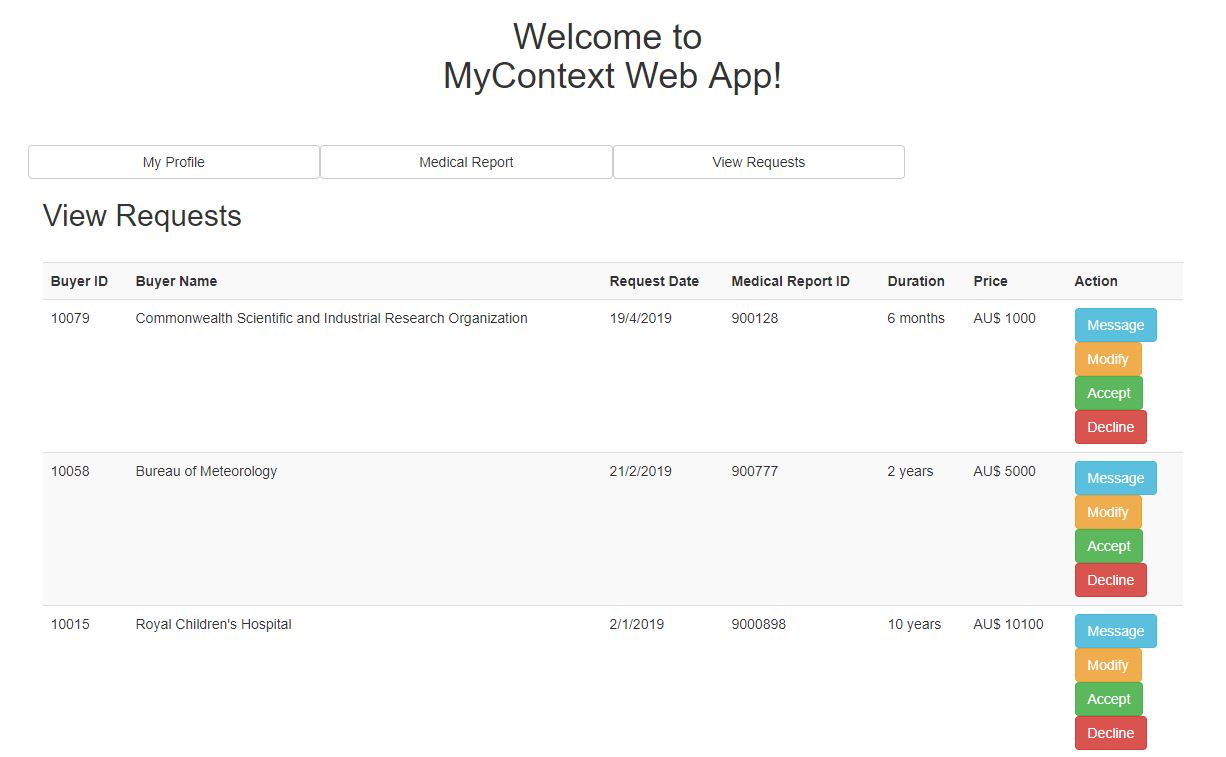


Figure : View Requests/Buyers Page

### Popup and Notification Feature

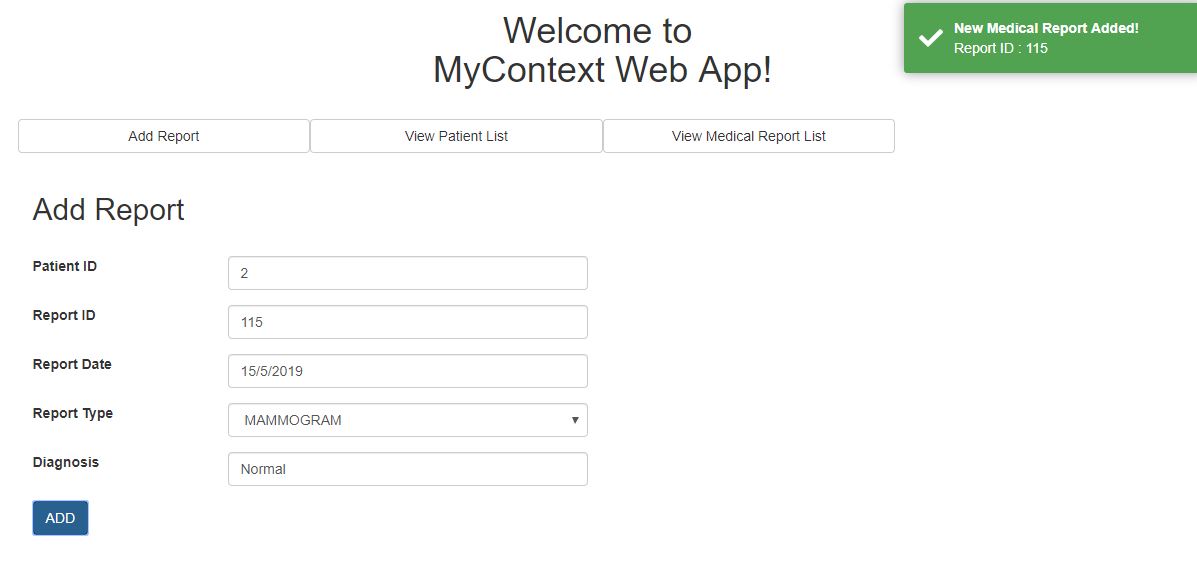


Figure : Notification of successful operation

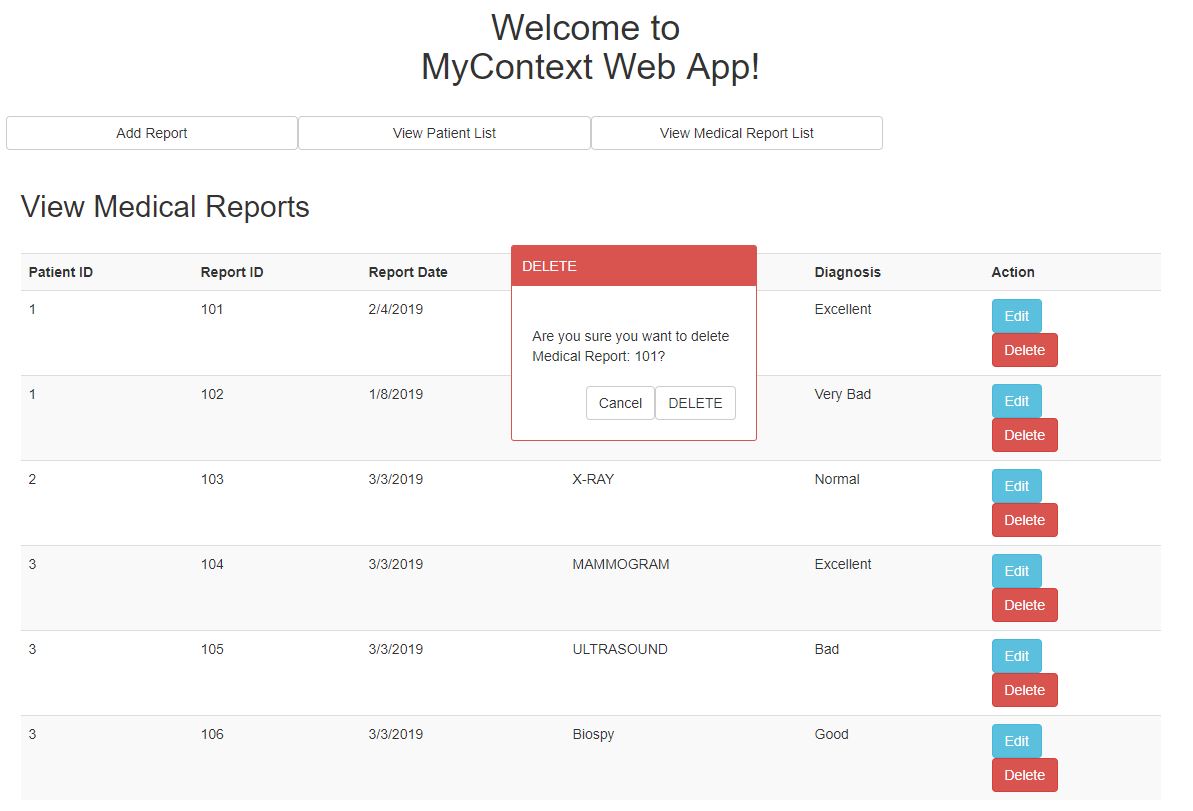


Figure : Prompt message for confirming user action

## Plans for Next Phase

1. Creating Buyer dashboard.
   1. Search and filter Medical Reports.
   2. Request to buy medical report to Patient.
   3. Purchase medical report from Patient.
2. Complete all UI pages.
3. Integrate Login and Registration Pages created by Saisri Mokkapati.
4. The app currently consists of services which supplies data from variables. Services must be updated to call web API using Angular HTTP Client module.

## References

<https://angular.io/guide/quickstart>

<https://loiane.com/2017/08/how-to-add-bootstrap-to-an-angular-cli-project/>

<https://github.com/scttcper/ngx-toastr>

<https://www.npmjs.com/package/ng2-popup>