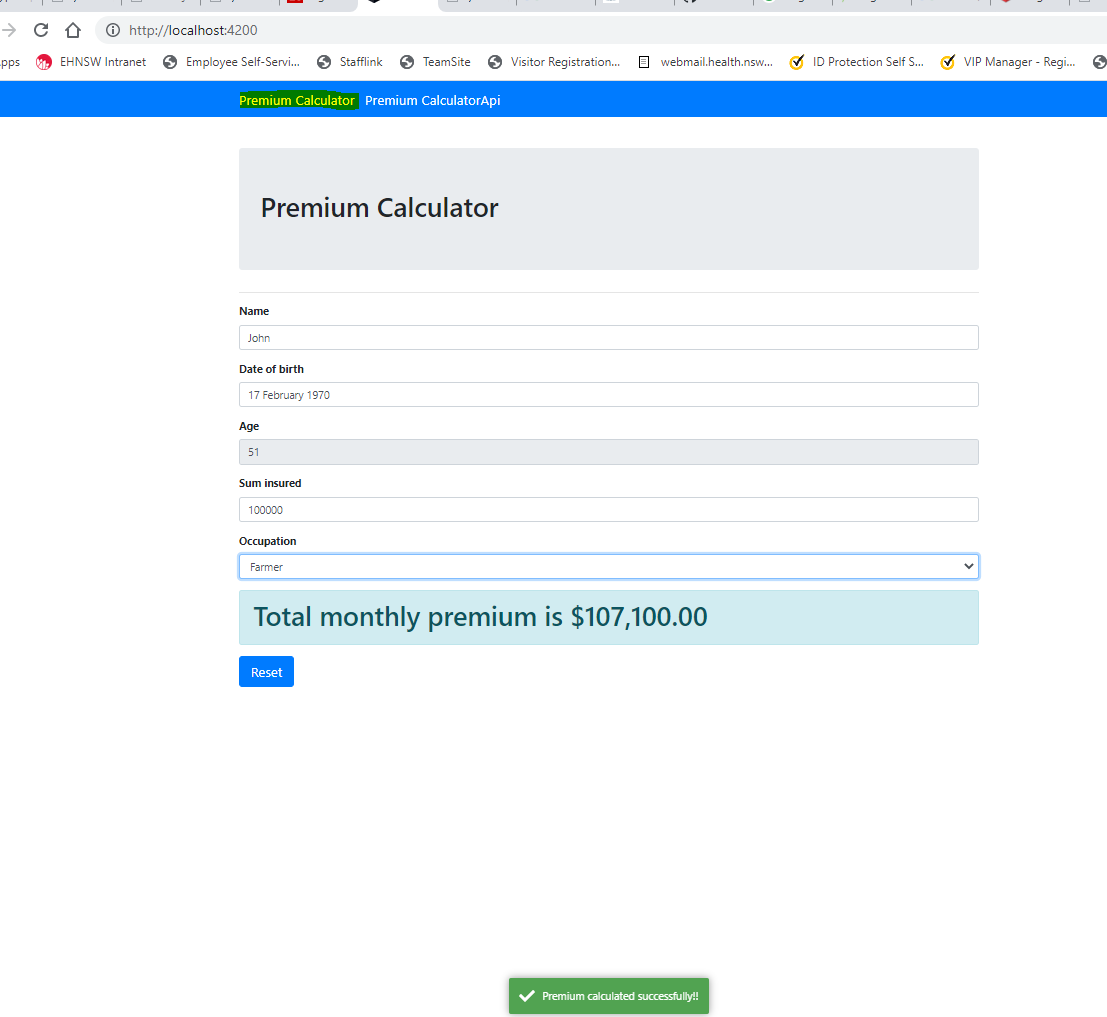
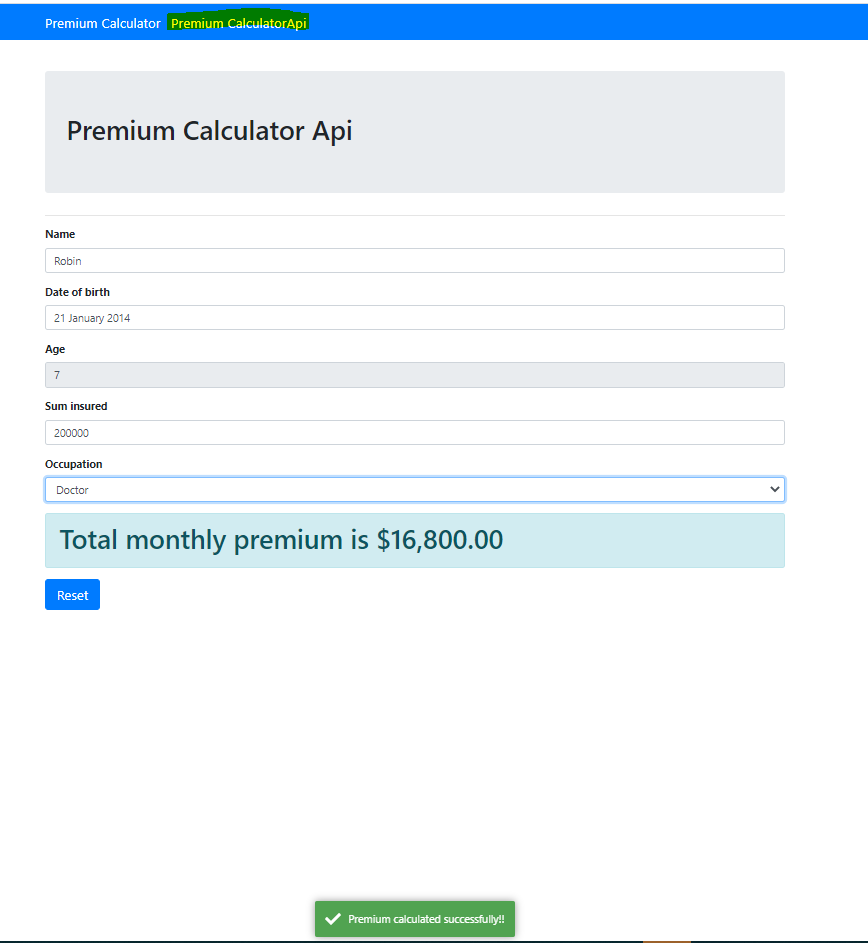
Premium Calculator

Application has been built with two approaches,

**First** Client side approach where calculation logic has been written in angular/typescript as per following screen.



**Second** FrontEnd used angular components to with reactive forms for validations and have created services to consume API and all logic resided in API which in response return calculated premium, this approach is more scalable and secure.



Validation has been implemented on both components with Reactive forms,

created custom @directives for validation of number only and alphabets only

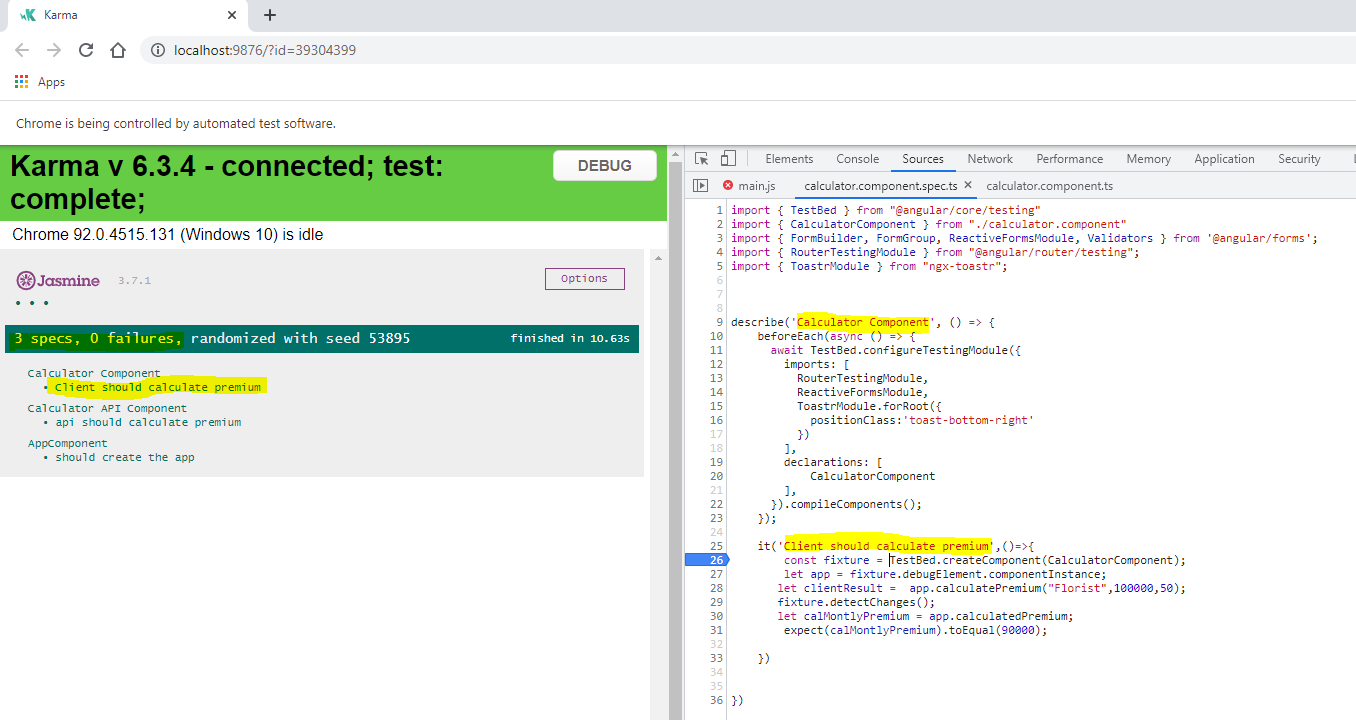
1. appNumberOnlyValidator
2. appAlphabetOnlyValidation

Added notification service for better UX



Added Test cases for Test Driven code for API component and Client component, Run tests with following command





**Technical Components**

Application build in two components API and Client.

API contains web api’s to handle http requests and contains Zip validation logics and CRUD operations

API Technology stack

* .Net5
* WebApi

Start API application by navigating to API folder and running following command in VS code terminal or Command prompt

“dotnet run watch”



Client (Angular 12) application is Frontend and consuming web api’s to perform validations

Client Technology stack

* Angular12
* Ngx-bootstrap
* Toastr

Angular application required node.js, npm, angular/cli installed, Start Client application by navigating to Client\Src\app folder and running following command in VS code terminal or Command prompt.

“npm start”

