**WeCare Application**

WeCare is an online Life Coaching application that helps its users to sign up and log in to seek the guidance of famous Life Coaches. Users can search for a Life Coach based on specialty and can book an appointment within 7 days. They can also see upcoming appointments and can reschedule or cancel the appointments. Similarly, Life Coaches can also sign up and login and can see their upcoming schedule.

1. **Home Screen**

This screen should render the home component of the application. Below view should be displayed.



The navigation bar should have brand icon and call us by default (before login).

On click of ‘Join as a Coach’ button Coach Signup Component should be loaded.

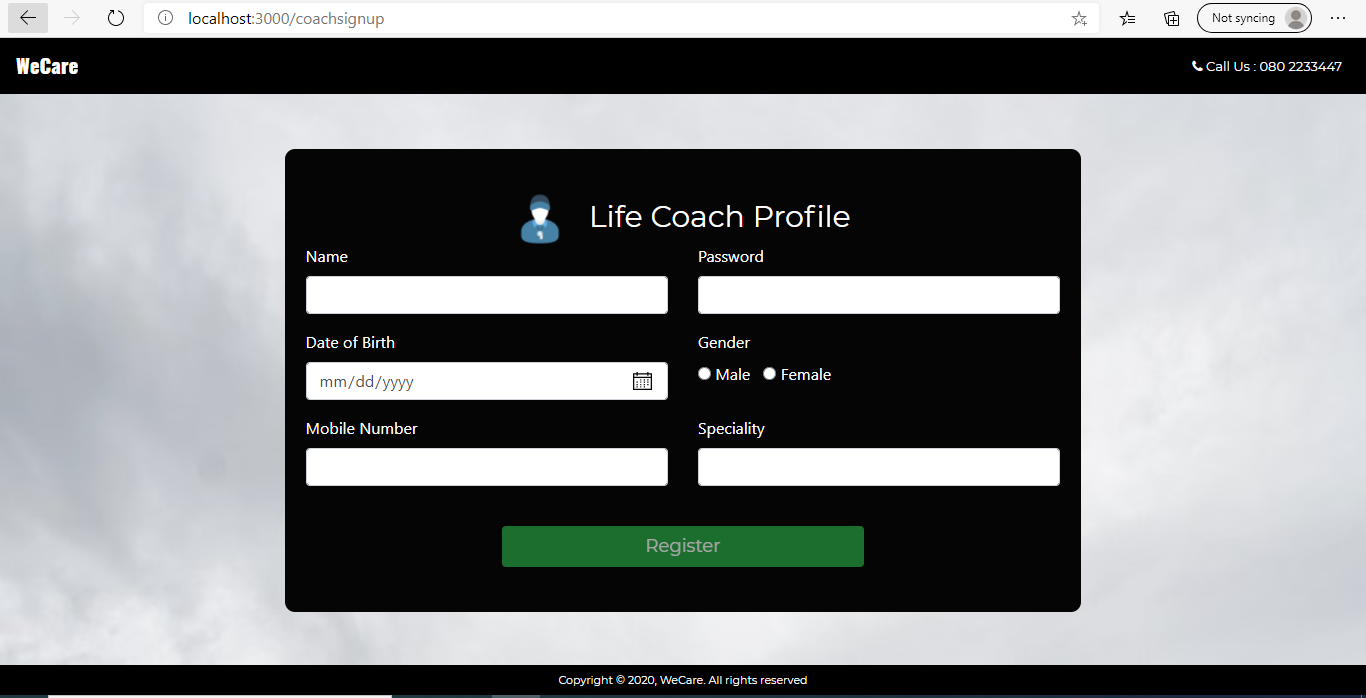
On click of ‘Login as a Coach’ button Coach Login Component should be loaded.

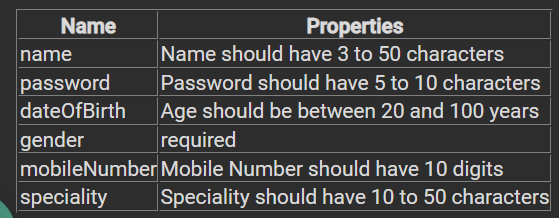
On click of ‘Join as a User’ button User Signup Component should be loaded.

On click of ‘Login as a User’ button User Login Component should be loaded.

1. **Coach Signup Screen**

This screen should render a Coach signup component for coaches to register into the application.

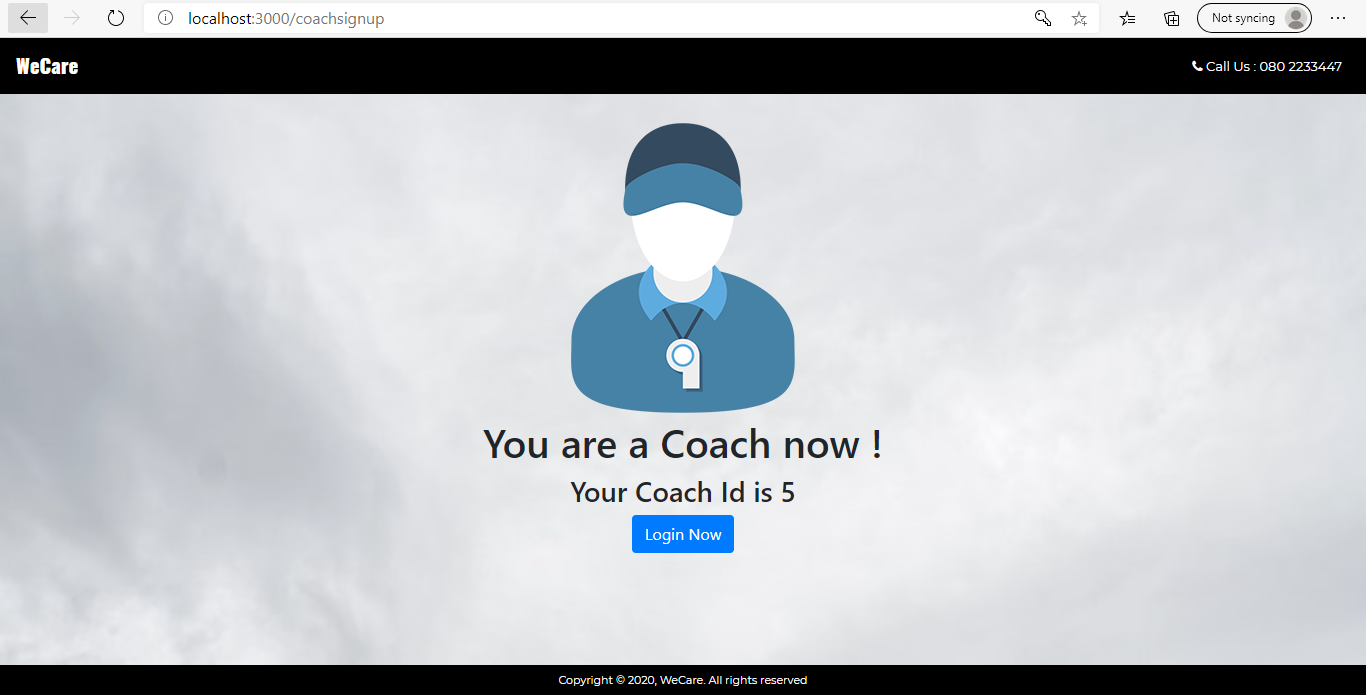




If any validation fails, display the error message as shown below:



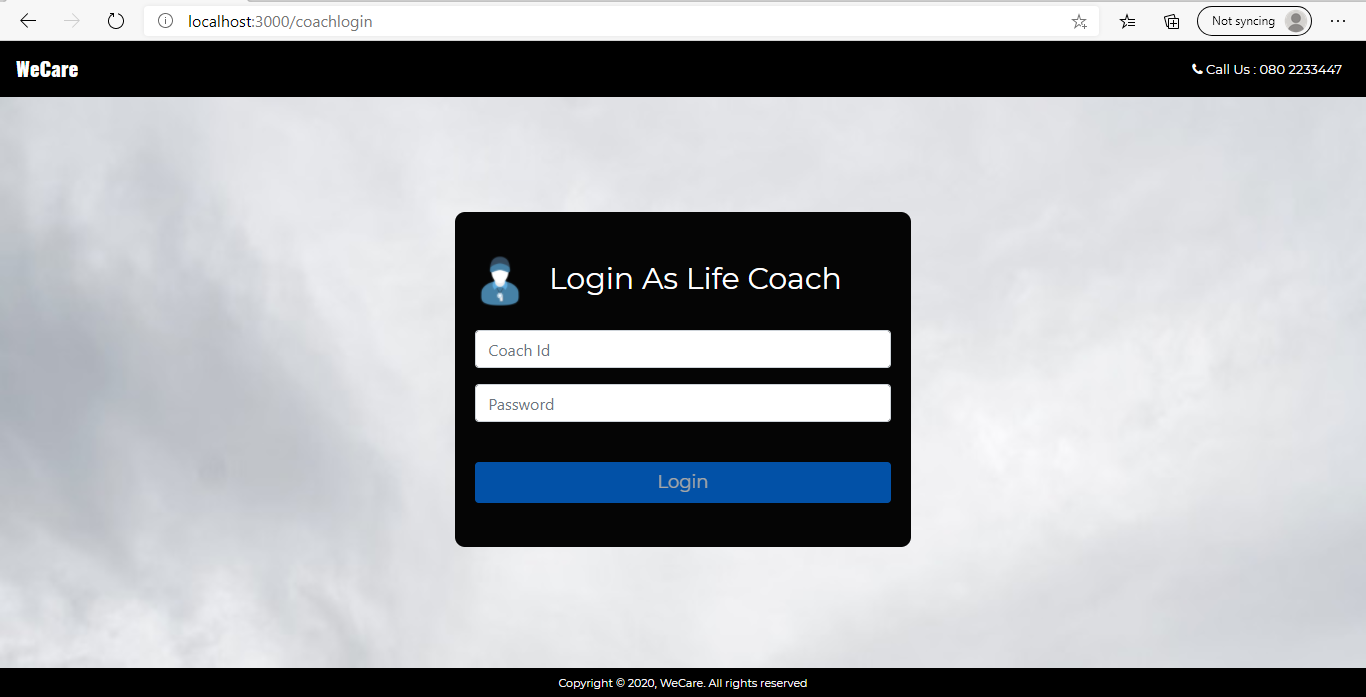
When the user clicks on register button after entering the values, send a post request to json server running at 8080 with path ‘/coaches’ and pass the collected data. If the registration is success make the below page rendered and code should be written in the same component.

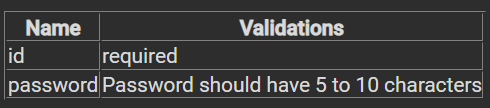


On click of ‘Login Now’ it should load coach login component.

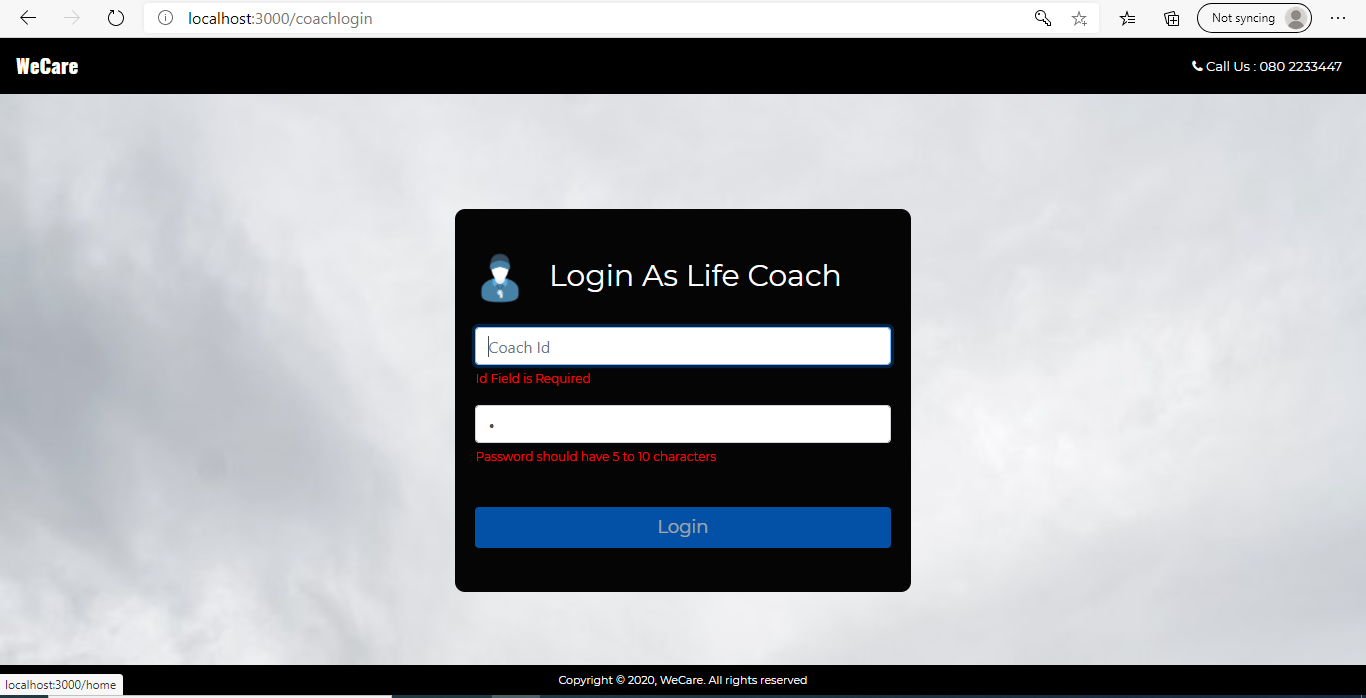
1. **Coach Login screen**

This screen should render a Coach login component for coaches to login into the application.

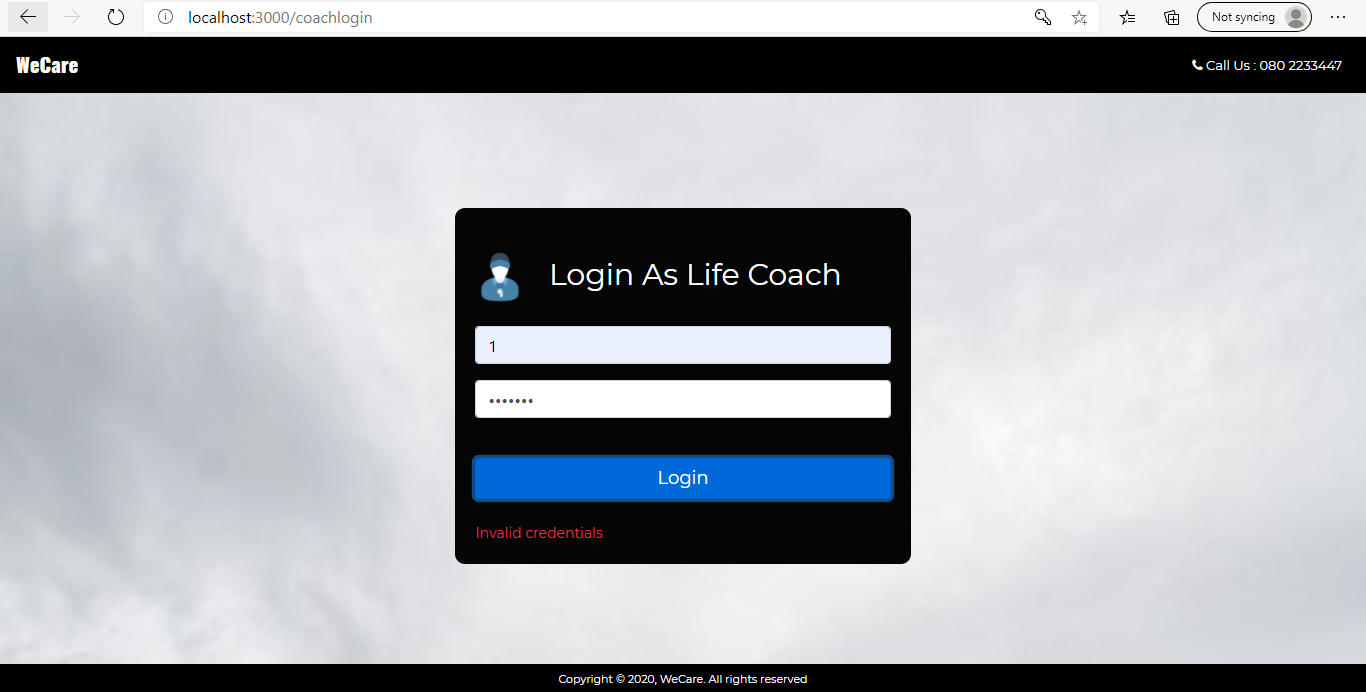




If any validation fails, display the error message as shown below:



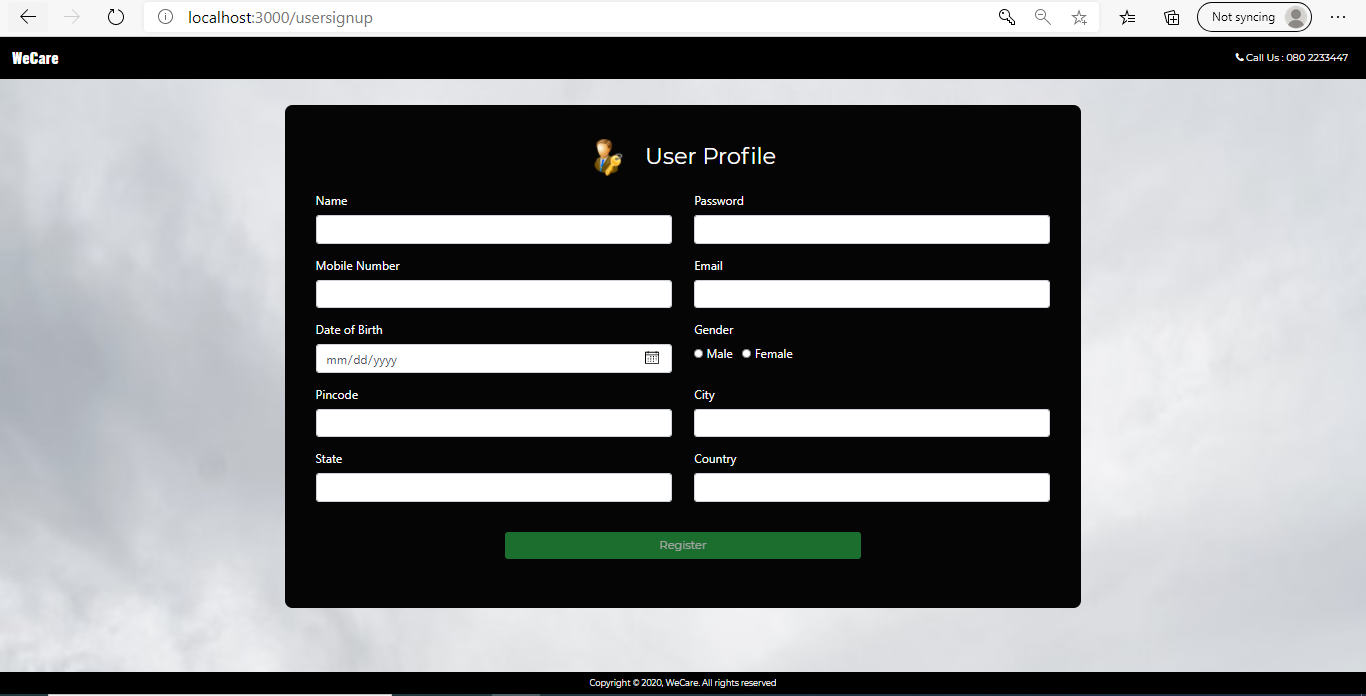
When the user clicks on login button after entering the values, send a get request to json server running at 8080 with path ‘/coaches?id='+entered id+'&password='+entered password’. If the coachId and password does not exist in coaches throw the below error.

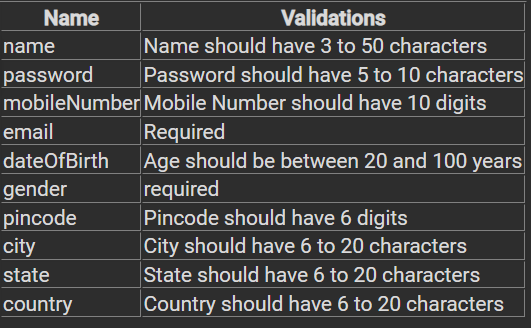


On successful login make Coach home component loaded.

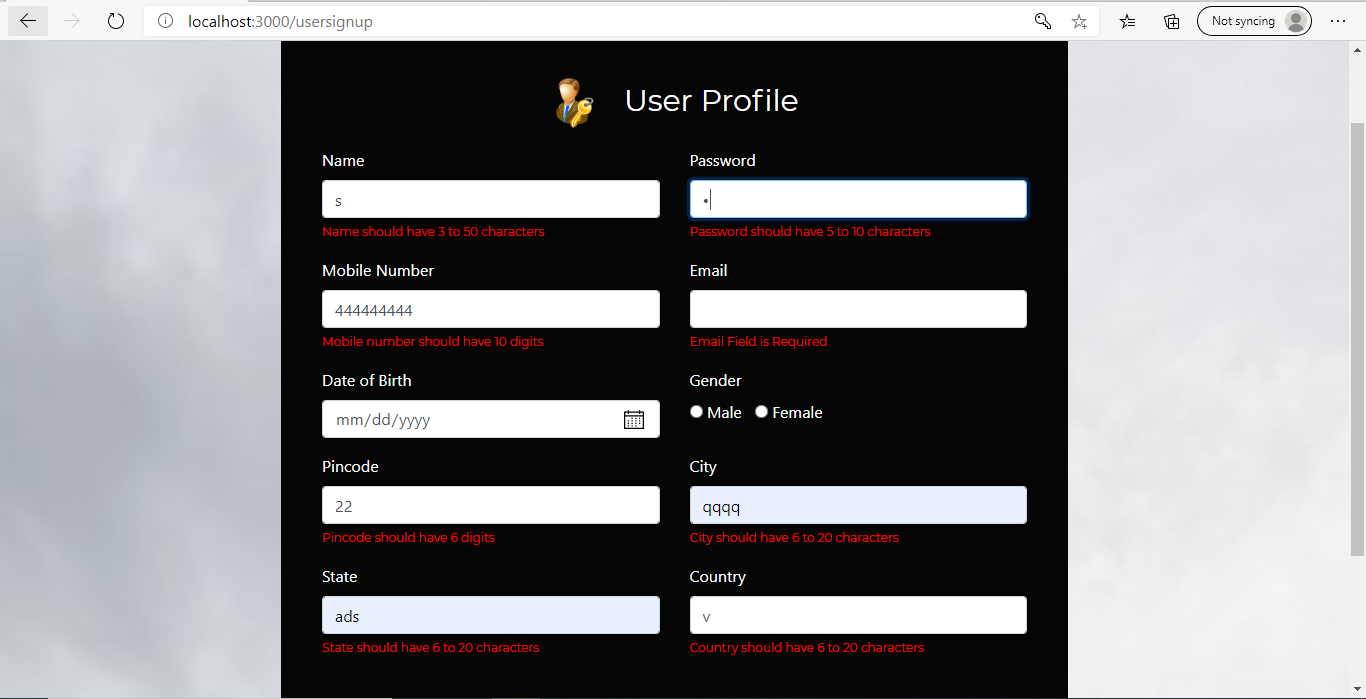
1. **User Signup screen**

This screen should render a User signup component for user to register into the application.

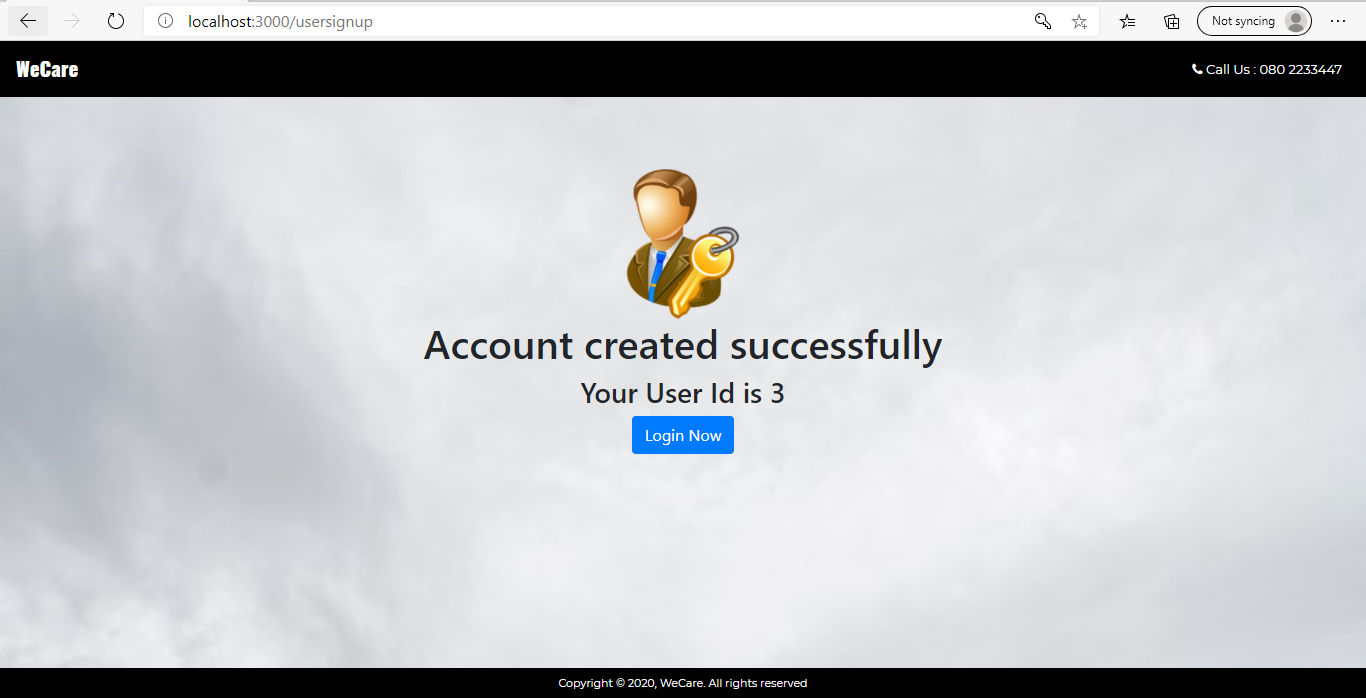




If any validation fails, display the error message as shown below:



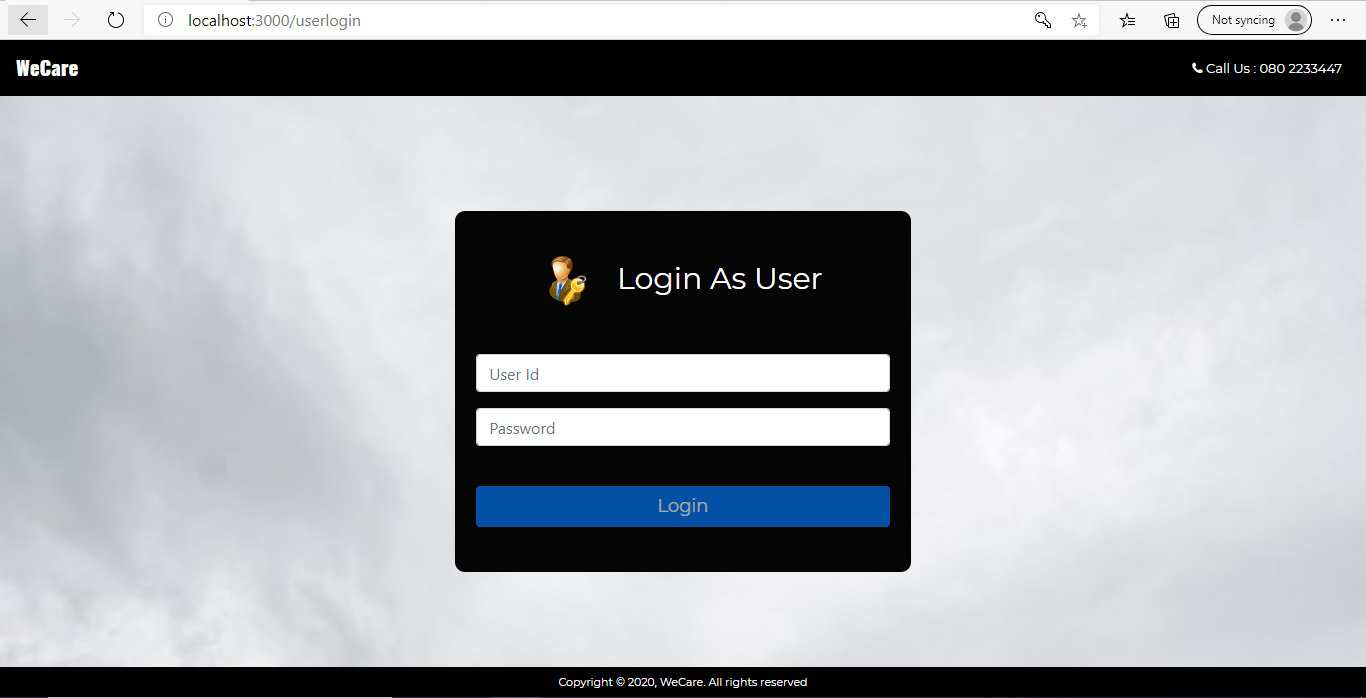
When the user clicks on register button after entering the values send a post request to json server running at 8080 with path ‘/users’ and pass the collected data. If the registration is success make the below page rendered and code should be written in the same component.

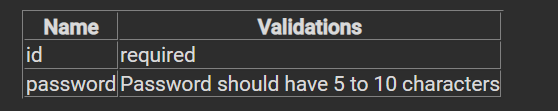


On click of ‘Login Now’ it should load User login component.

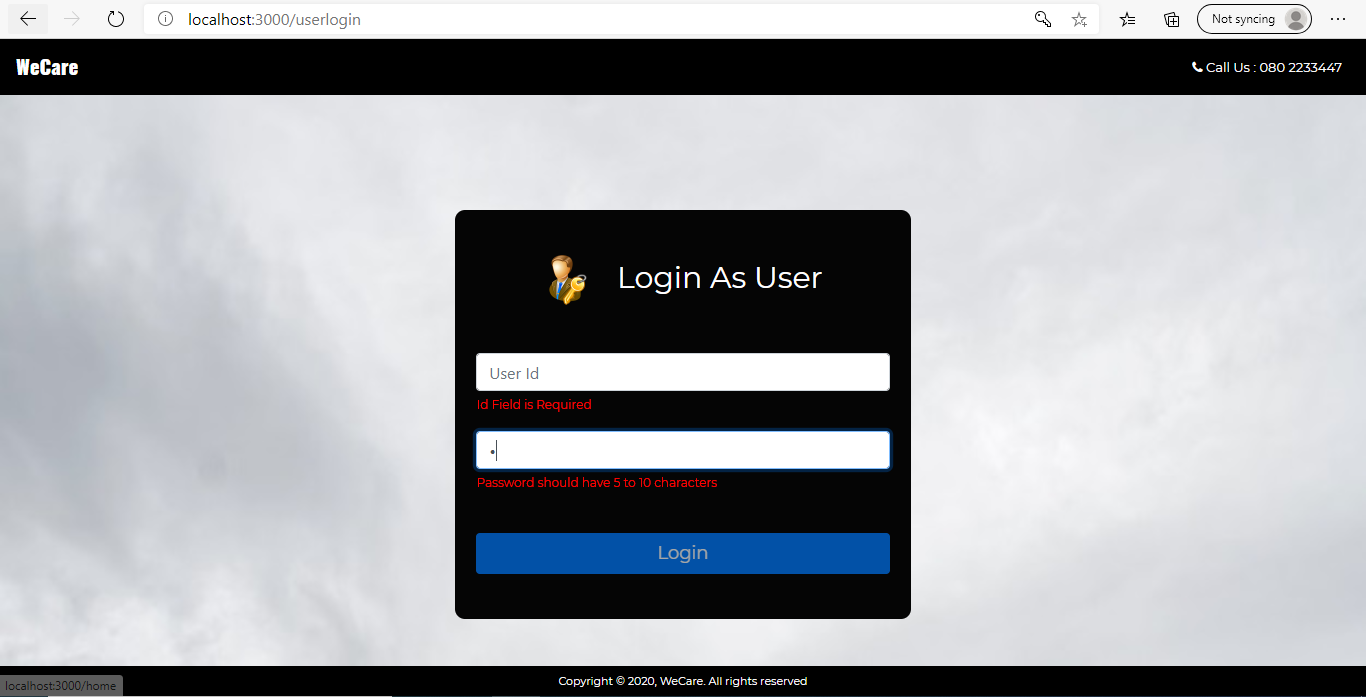
1. **User login screen**

This screen should render a User login component for users to login into the application.

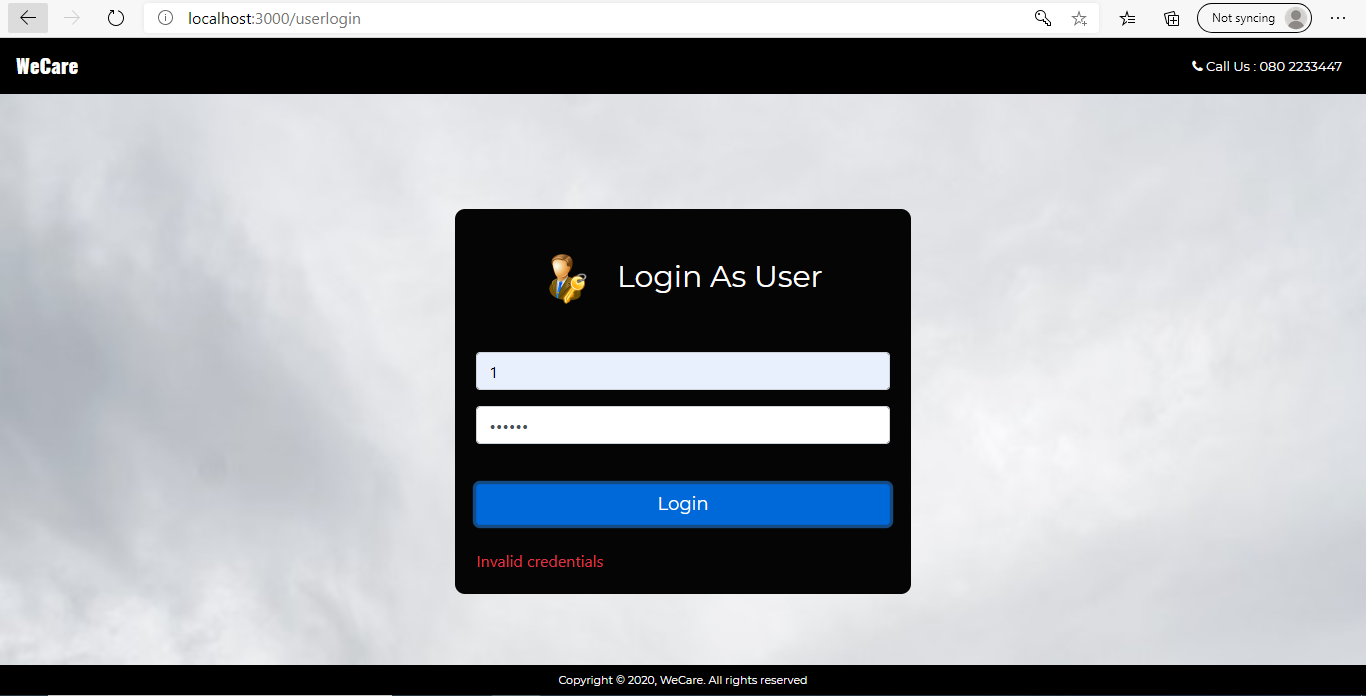




If any validation fails, display the error message as shown below:



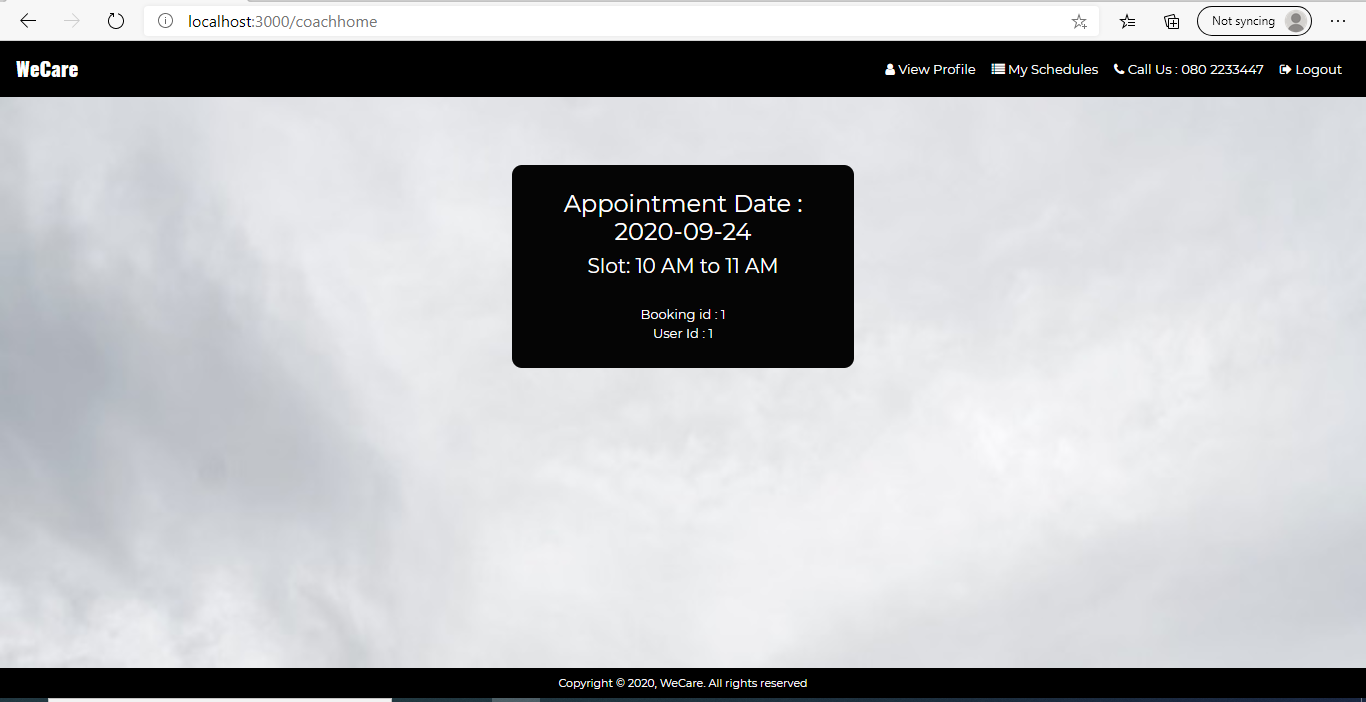
When the user clicks on login button after entering the values, send a get request to json server running at 8080 with path ‘/users/?id='+entered id+'&password='+entered password. If the user is not registered show the error message as shown below:



On successful login make User home component loaded.

1. **Coach Home Screen**

This screen should render the Coach home component. On load of this component it should render the below screen. Navbar should be updated as shown below in screenshot.



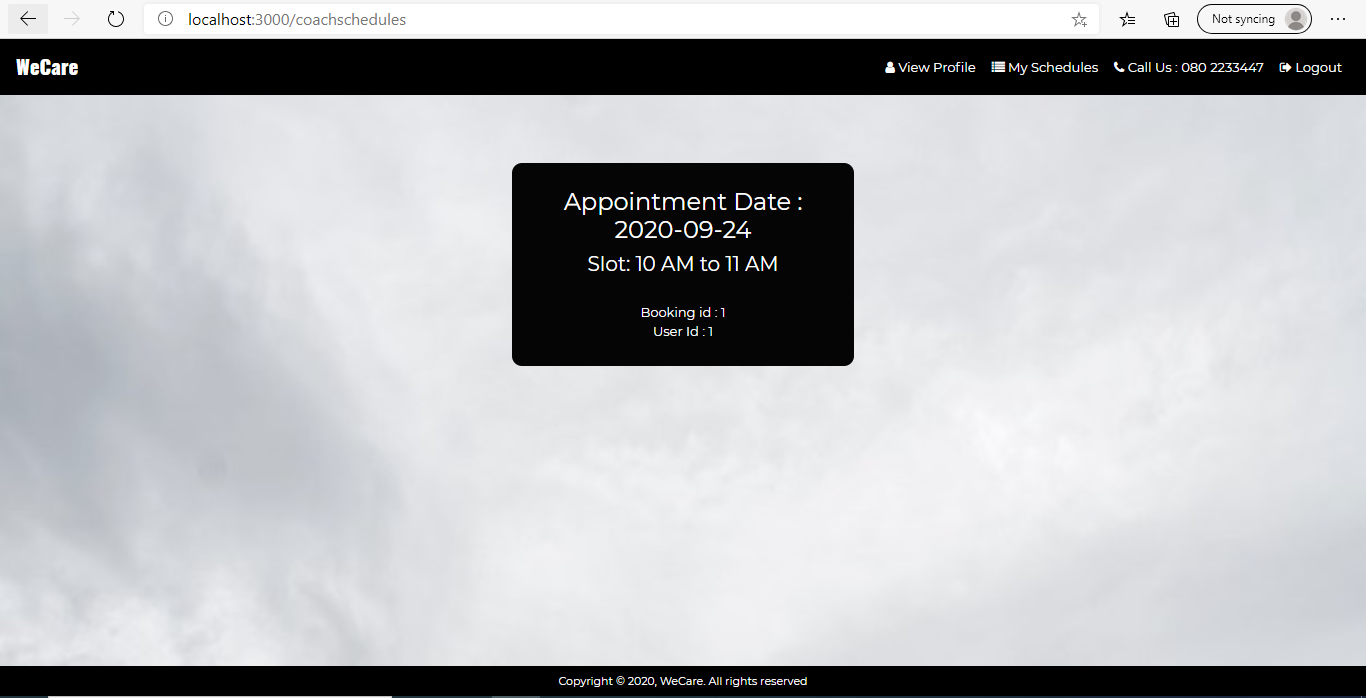
On load of this component,

1. Send a get request to json server running at 8080 with path ‘/bookings’ and populate appointment details with the success return value.

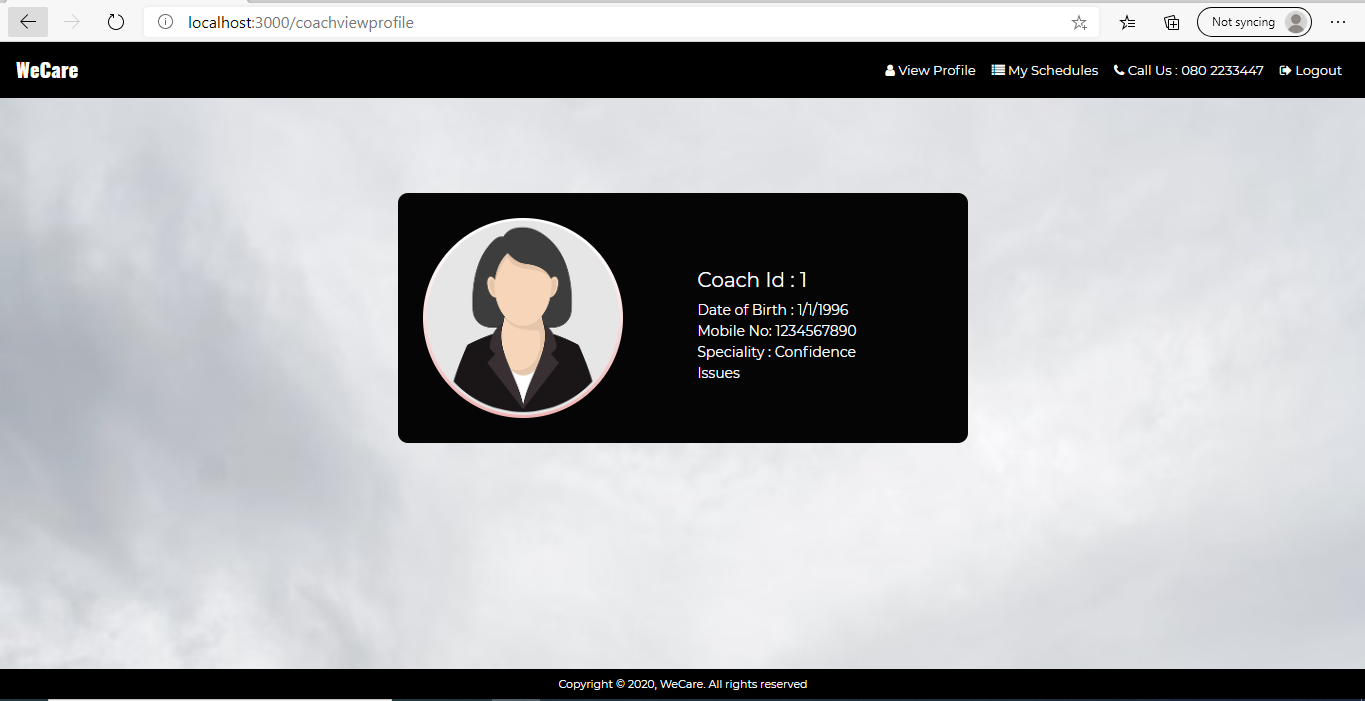
2. Send another get request to json server running at 8080 with path ‘/coaches/:coachId’ and populate coach details with the success return value.

On clicking navbar links it should show schedule details and coach details appropriately as shown in the below screenshots.

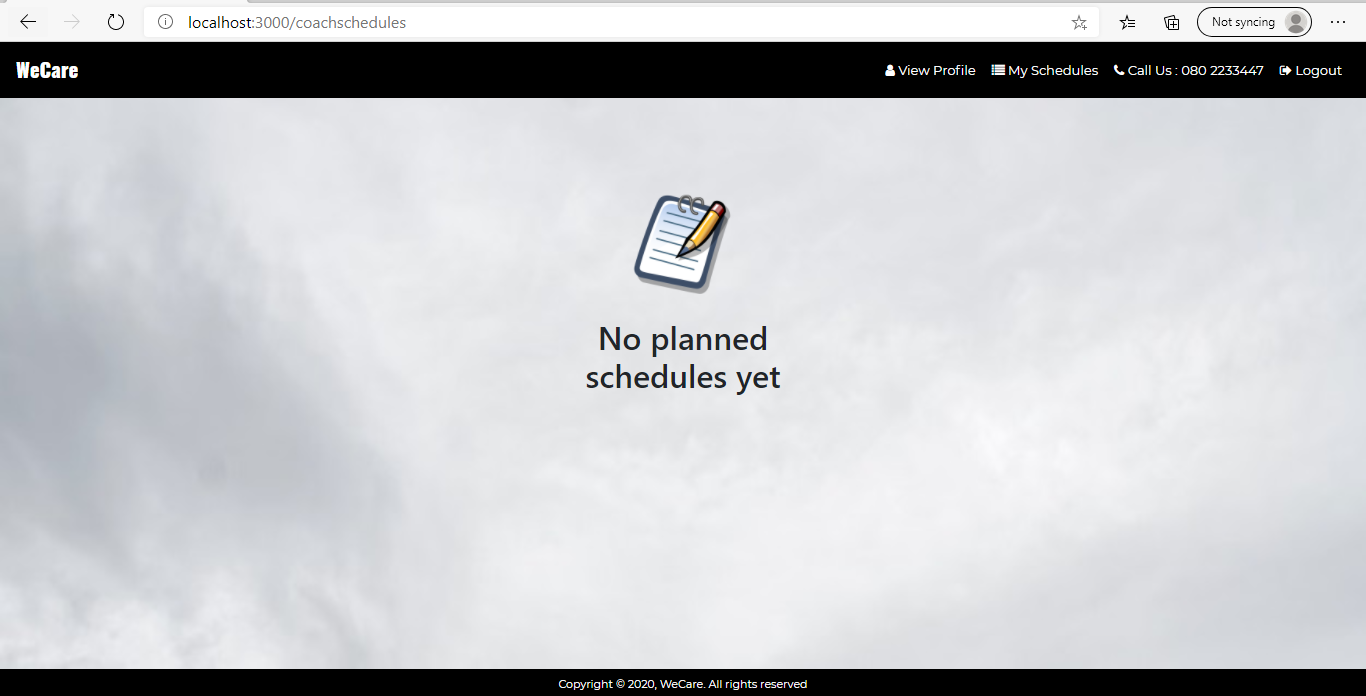
On clicking My Schedules link it should display the below page:



On clicking View Profile link it should display the below page:

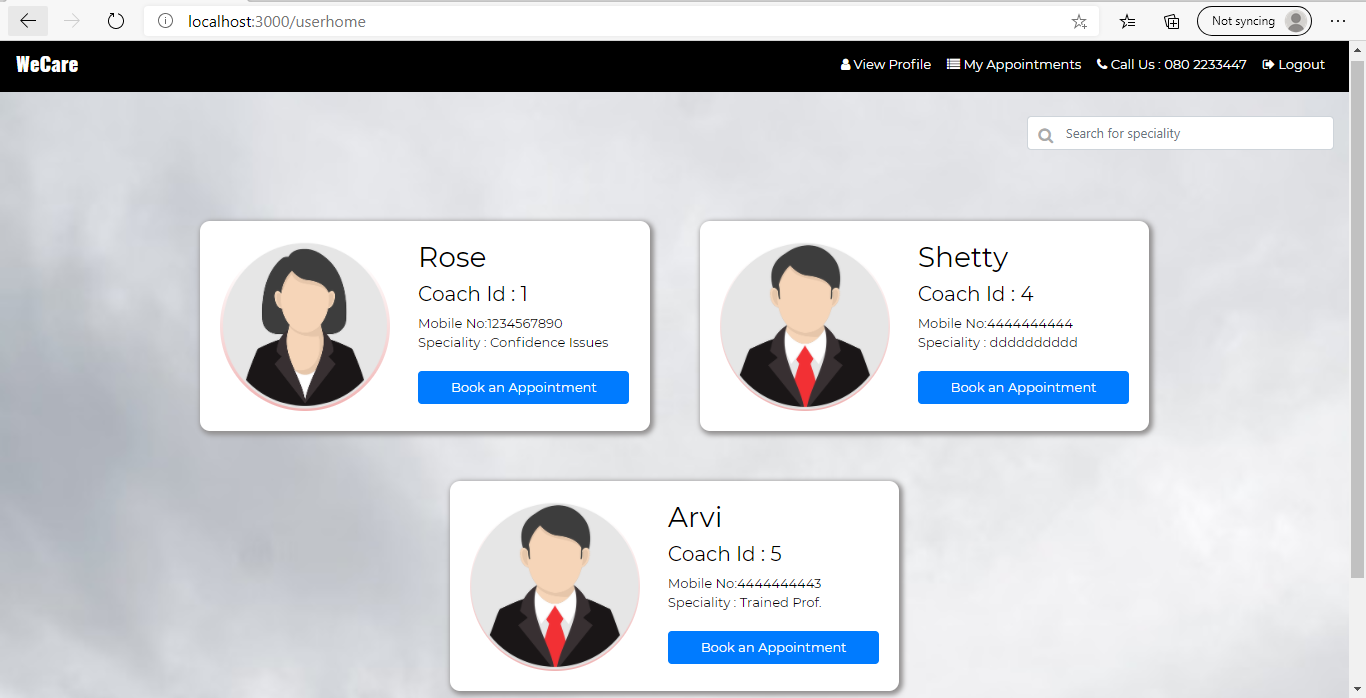


On load of /coachhome and /coachschedules it should show schedule details initially if any. If not, schedules fixed yet, the below display should be shown:



1. **User Home Screen**

This screen should render the User home component. On load of this component it should render the below page. Navbar should be updated as shown below in screenshot.



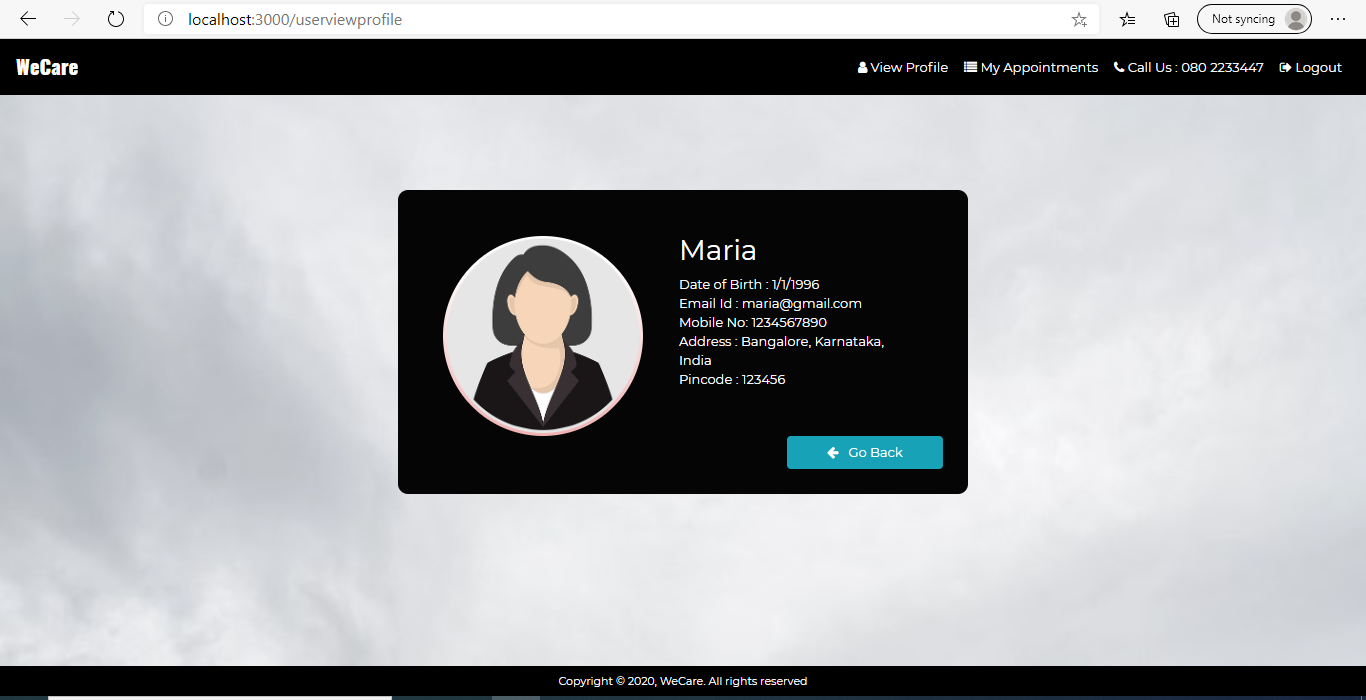
On load of this component,

Send a get request to json server running at 8080 with path ‘/coaches’ and populate coach array with the success return value.

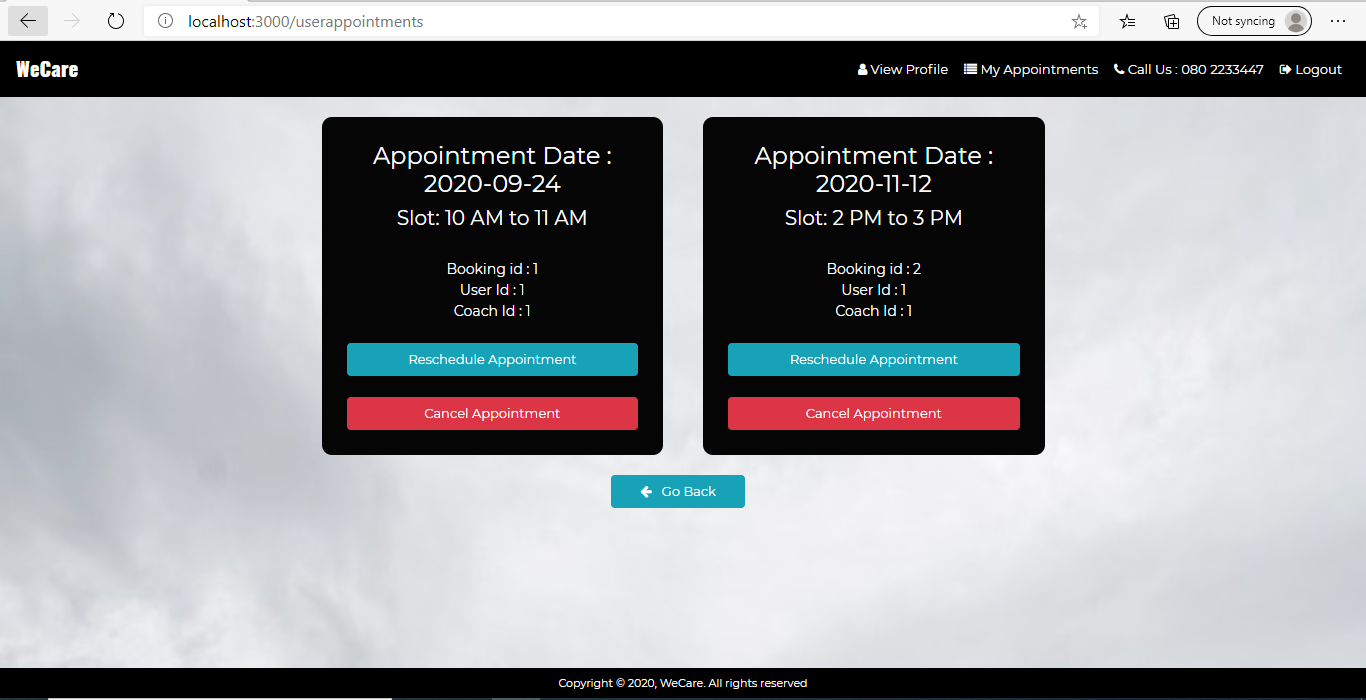
Send a get request to json server running at 8080 with path ‘/users/:userId’ and populate user details with the success return value.

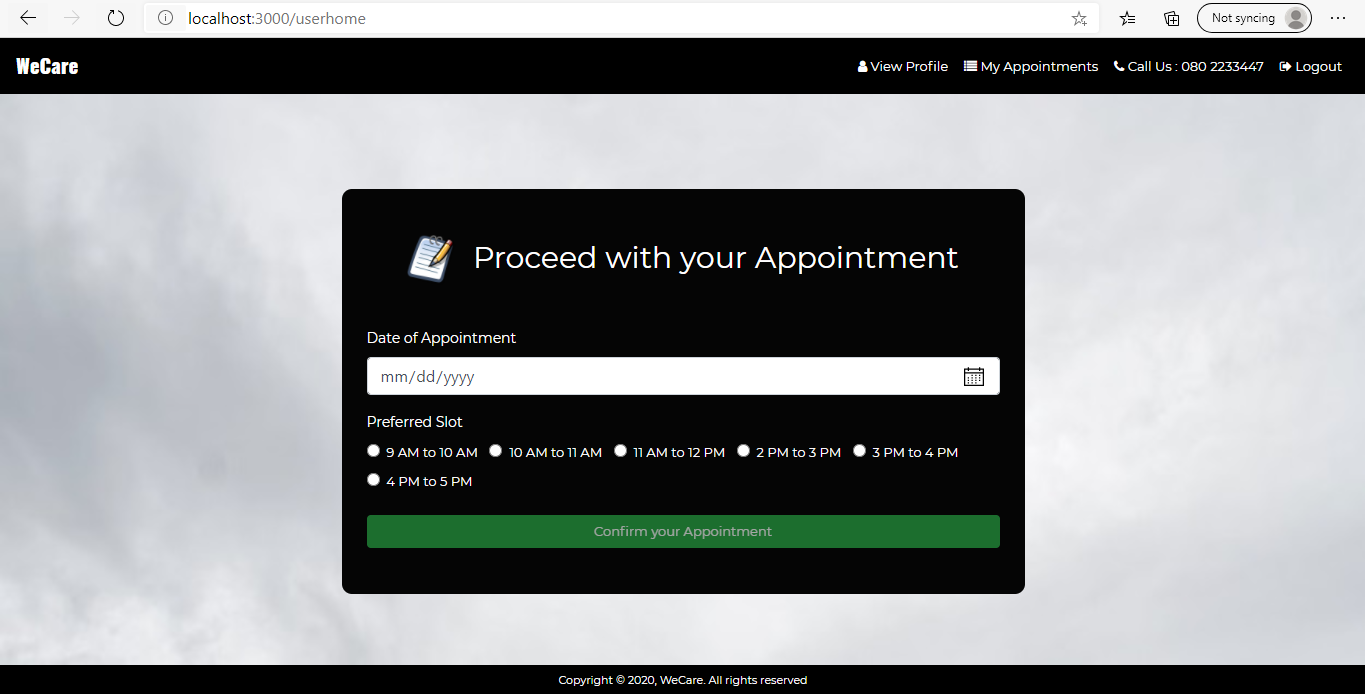
Send a get request to json server running at 8080 with path ‘/bookings’ and populate appointment details with the success return value.

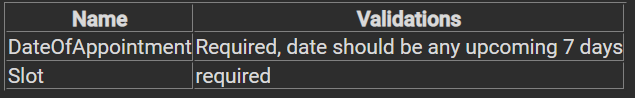
On clicking navbar links it should show appointment details and user details appropriately. On load of this component (with URL:/userhome) it should show coach array initially as shown above.

On clicking View Profile link it should display the below page:

On clicking My Appointments link it should display all the appointments made by the user as shown below.

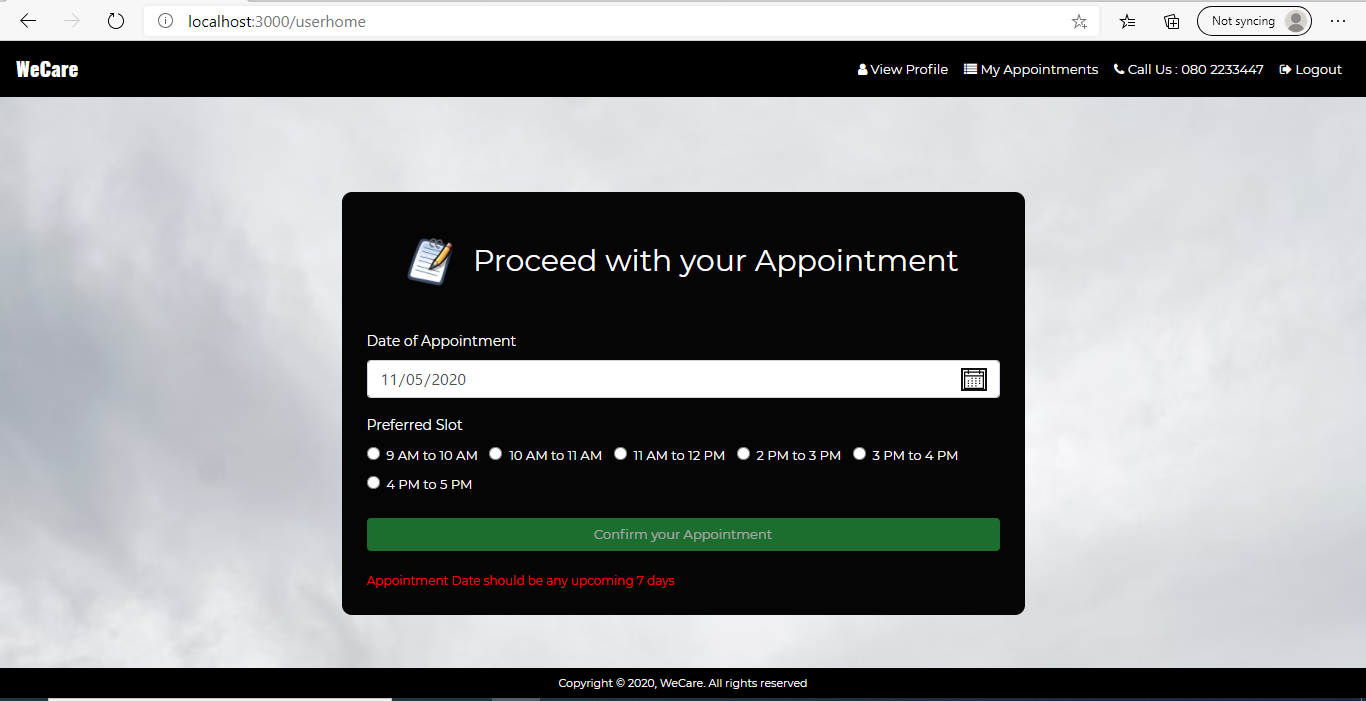


On click of Book Appointment button on any coach Card below page should be shown:

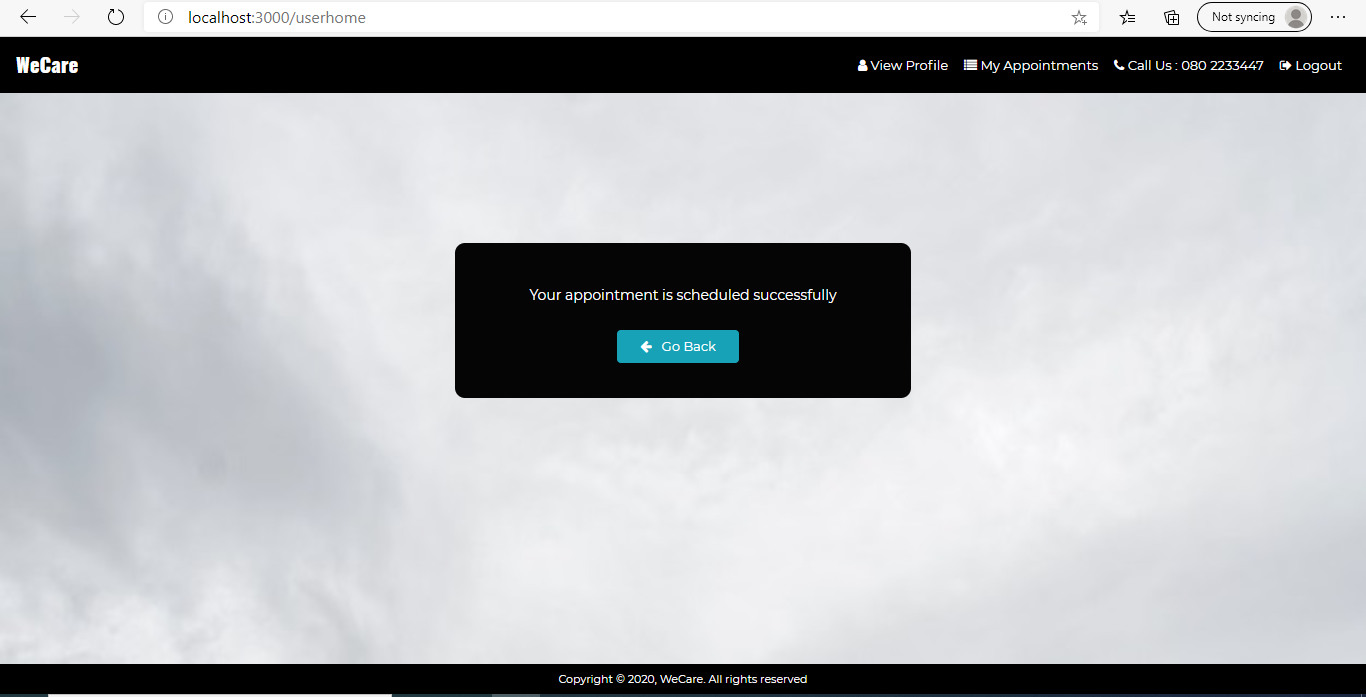


Button will be disabled until all validations are satisfied. On click of Confirm your Appointment it should make a post request at ‘/bookings’ with data collected (along with userId and coachId)

If any validation fails, display the error message as shown below:

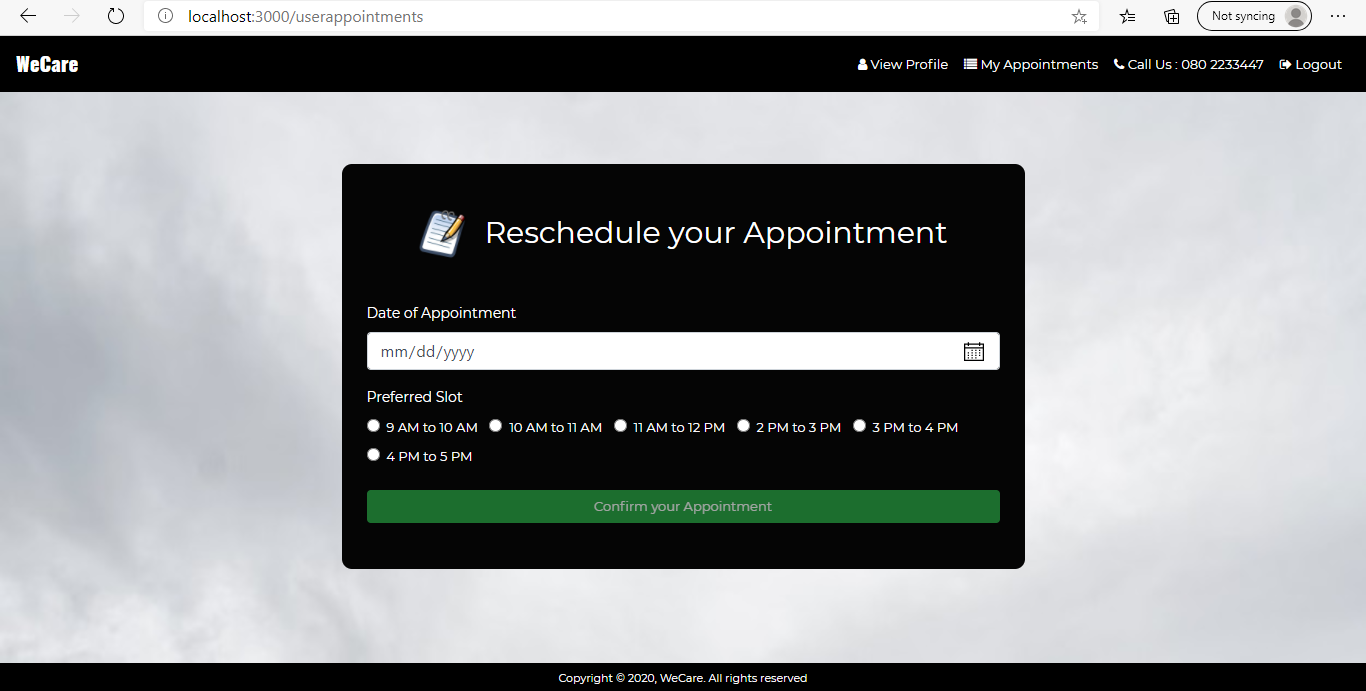


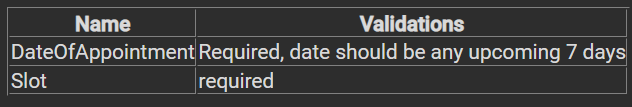
On successful making of any appointment show the below page:



On clicking the go back button, it should navigate to /userhome page where it displays the list of coaches.

On click of Reschedule Appointment Button on the /userappointment display the below page:



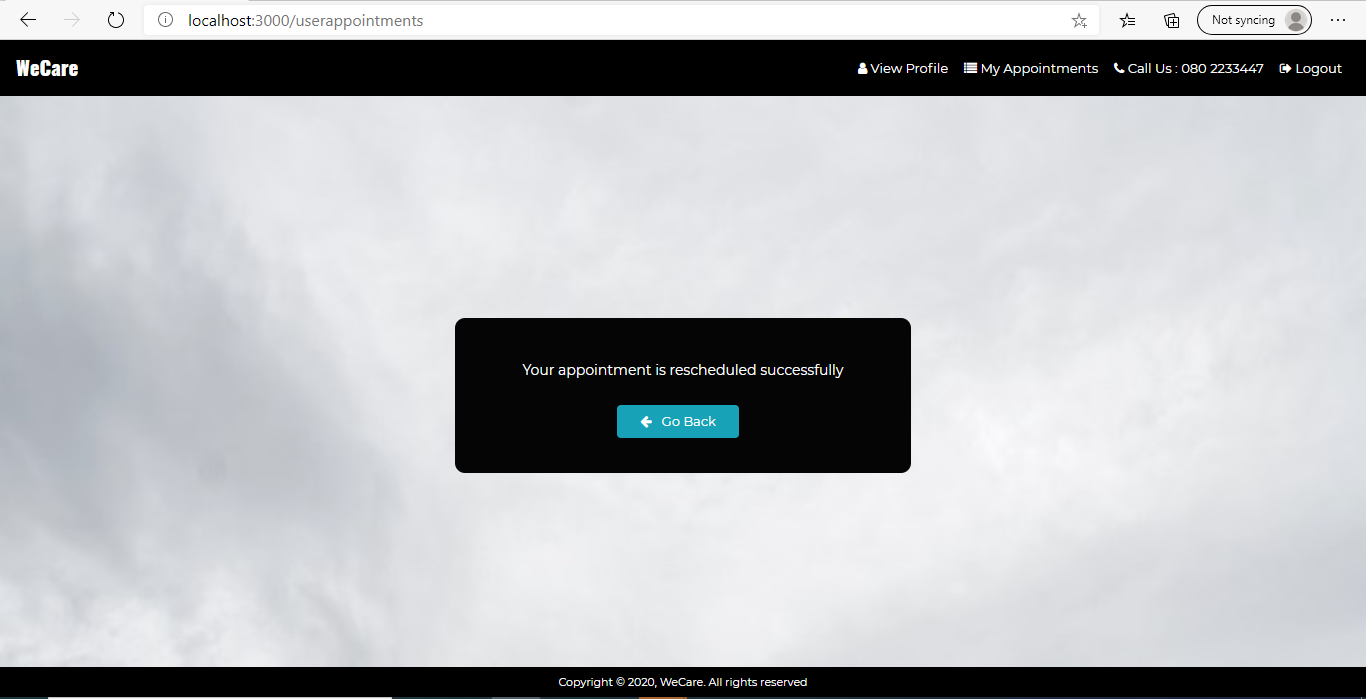


Button will be disabled until all validations are satisfied. On click of Confirm your Appointment it should make a patch request at ‘bookings/:bookingId’ with data collected

If any validation fails, display the error message as shown below:

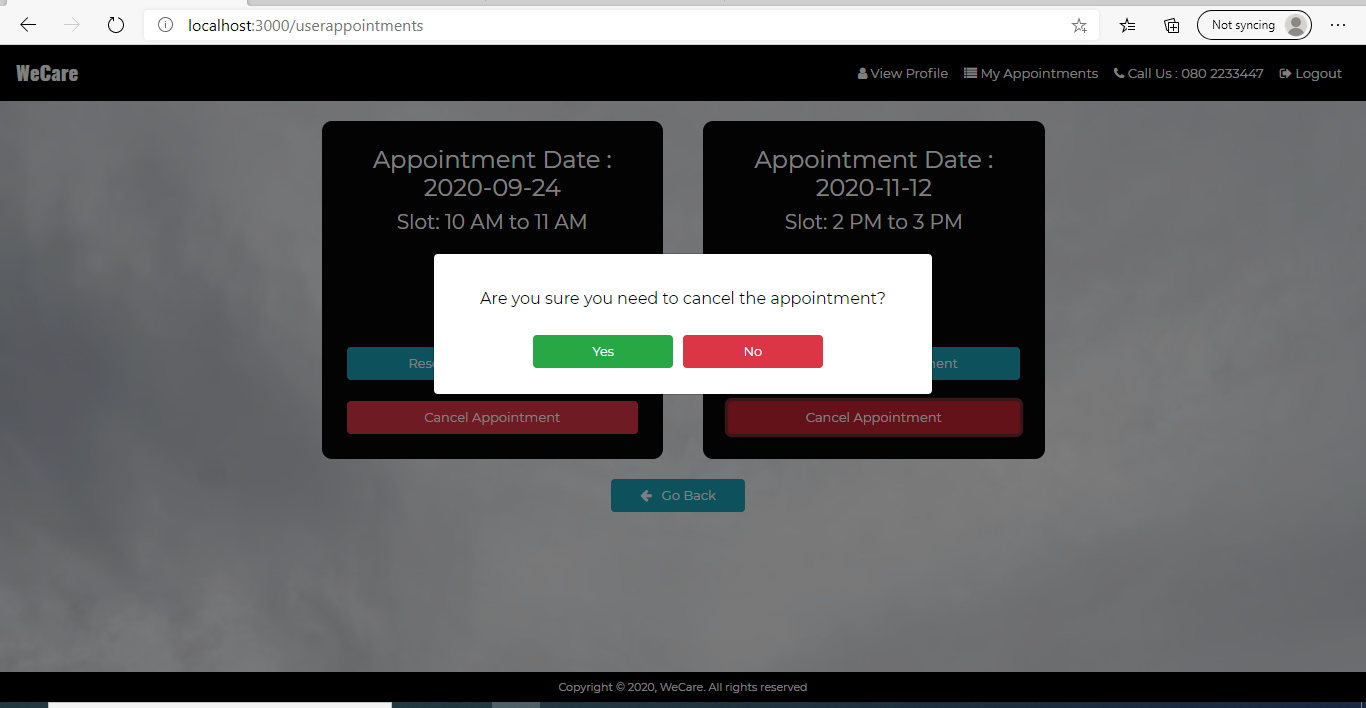


On successful rescheduling show the below page:

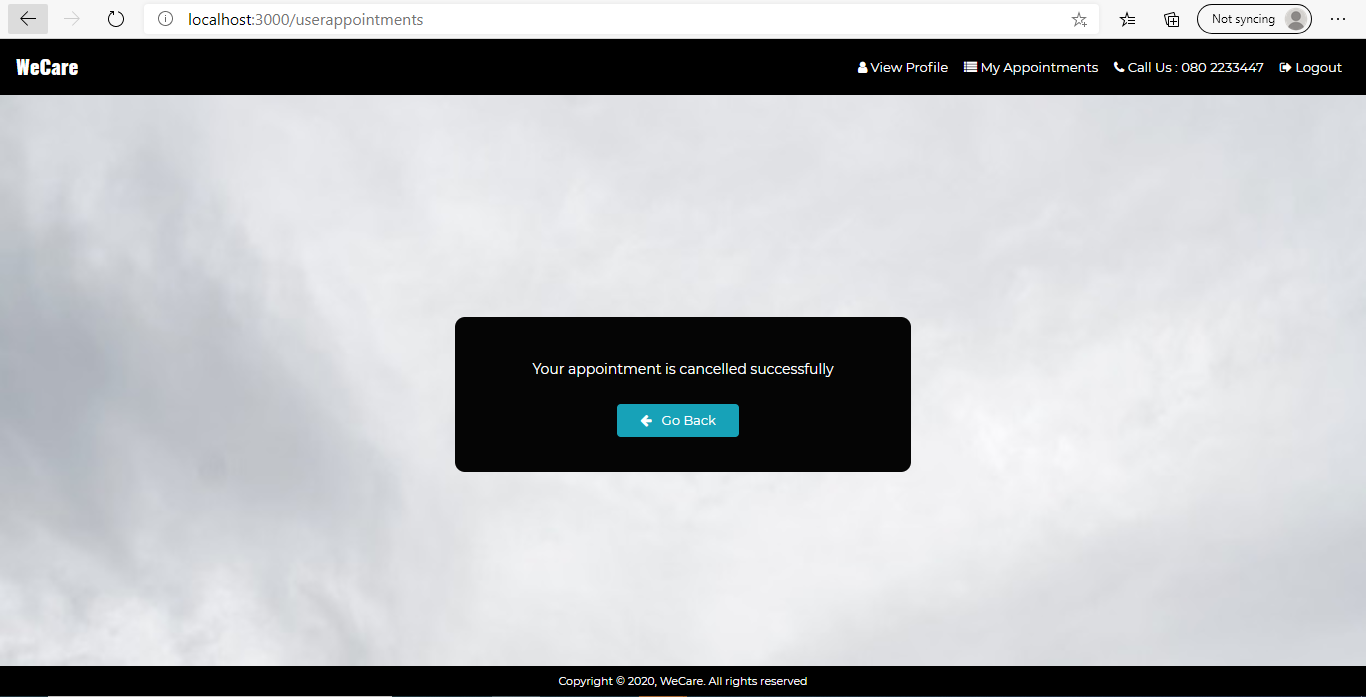


On clicking the go back button, it should navigate to /userhome page where it displays the list of coaches.

On click of cancel Appointment Button on /userappointment, it should invoke a model which is shown in the below screenshot. When user click the yes button in the model it should make a delete request at ‘/bookings/:bookingId’



On successful deletion of appointment show the below page:



On clicking the go back button, it should navigate to /userhome page where it displays the list of coaches.