CSE3002- INTERNET AND WEB PROGRAMMING

PROJECT REVIEW- 3

# WORKFORCE

## A Web Based Application for Providing Home Services


Submitted by:

S.R.PRAVEEN (19BCE0363)

GUNNAM MAHESH CHOWDARY (19BCB0139)


in partial fulfillment for the award of the degree of

**B. Tech**

**in**

**Computer Science Engineering**


VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Under the guidance of:

**Prof. Nalini N.**

## ABSTRACT:

Our primary goal is to build a website that acts as a platform which enables skilled and experienced professionals to connect with users looking for specific services. Because of the current pandemic, we have to be always very cautious. But there are certain needs and requirements in our life that can't be ignored**.** These needs and requirements just don't vanish because of the pandemic or take into consideration some of the strict rules imposed by the government on us. Many people have lost their jobs thanks to the pandemic. Our idea offers these people fresh hope. Anyone can register and list the service they offer in our website. We display it to our users who search for these services. Our website charges no money from the service provider. It just acts as a platform through which service providers can list all the services they offer to a wider audience and customers can search for quality professionals who are willing to offer the particular service our customer is looking for.

The services offered by the professionals registered with WorkForce website are all home services and these services are all offered on demand. We identify and display professionals who are closest to the users' requirements and also currently online at the time of users' searching for a service in WorkForce. From the list of registered professionals displayed on a users' query for a service, the user can choose any professional he/she wants too.


## INTRODUCTION:

Our primary goal is to build a website that acts as a platform which enables skilled and experienced professionals to connect with users looking for specific services. Because of the pandemic, we are forced to stay in our home for our own safety. But there are some necessities that can't be completely ignored. Certain needs and requirements are there in everyone's day to day life. These needs and requirements just don't vanish because of the pandemic or take into consideration some of the strict rules imposed by the government on us. So, how is it possible to balance our needs and also ensure our safety? WorkForce. WorkForce offers a platform that assists skilled and experienced professionals to connect with users looking for specific services. No one wants to expose themselves to the corona virus. But as mentioned earlier there are certain things that cannot be ignored. In these critical conditions we are living, where we can't afford to take the risk of getting exposed to covid19 at any cost, it's robbed us all of having small simple things like a head massage, beauty treatments to name a few.

How do we get a haircut? What if you want a head massage? You want to get your walls done, where can you search for a painter? Your electrical appliances broke, how do you repair it and who do you call? How can you get all these things done immediately from your home? Workforce. Here in WorkForce, there are registered professionals who offer these services on-demand. The customers do need to do absolutely nothing once they have booked a service. The professional arrives at the customer's door with all the necessary tools/things required to complete the work. The services offered by the professionals

registered with our website are all home services. There is no need to worry about the quality of the service we are providing, every registered professional in our website has years of experience in their profession and known for their work in their locality. A distinct feature provided by WorkForce is that you, our customers can choose the service provider i.e, a customer can choose any registered professional providing the service listed in our website (under customer's area).

Suppose you are in a hurry. You, our customer wants something done immediately like you want to get your hair done in the next 40 minutes/ you have a leaky water pipe. There is a high chance that you won't be able to get your hair done in the next 40 minutes/ find a plumber to fix your leaky pipe immediately. Who do you go to? You can come to us, just register in our website to access the services provided by professionals registered in our website. On searching for a service, the customer is only shown professionals who are currently available online near their residence. This feature helps customers who are looking for immediate service. Any user can register as a professional by entering the details of their service in the "Register as a professional" form. Then, our admin decides whether to accept/deny users' request (who wants to register as a service provider in our website). WorkForce charge no money in offering these services. The customers pay the service fee to their service provider directly. WorkForce only displays the pricings (as given by professional offering that service) for the services offered.

## INNOVATIVE IDEA:

There are some websites such as URBAN COMPANY which provides similar kind of service, but the problem we had identified with them is that only their employees would be providing such services, also employees need to travel to the customers place which there by increases the cost to the end customer, so as to overcome this problem we have come up with an idea where anyone who is having the required skill can join WORKFORCE and provide their service, to reduce the cost much further WORKFORCE displays the professionals residing in the near-by area of the customer and also available currently.

A distinct feature provided by WorkForce is that our customers can choose the service provider i.e, a customer can choose any registered professional providing the service listed in our website (under customer's area).

# TECHNOLOGIES USED:

## Frontend:

### VueJS:

In developing our website “WorkForce”, we have used VueJS as our frontend javascript framework. It is an open-source progressive JavaScript framework utilized in developing interactive web interfaces. Using VueJS enormously simplifies web development (one of the best javascript framework in existence in doing so). VueJS concentrates on the view layer. Without any problems, VueJS can be easily integrated into big projects for front-end development.

### Vue-Router:

With the assistance of Vue-Router library in our project, navigation between pages is performed without refreshing the page every time when it is loaded.

### Axios:

Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser and it supports the Promise API that is native to JS ES6. It can be used intercept HTTP requests and responses and enables client-side protection against XSRF. It also has the ability to cancel requests.

### Sweet Alert:

SweetAlert is a library to customize alerts in our websites, applications and games. It enables the user to change it with a standard JavaScript button. The user can add a new button to it, also change the button’s text, background colour of the button, and add additional alerts that depend upon the user's click.

## Backend:

### Node JS:

Node.js is an open source server environment. Node.js allows users to run JavaScript on the server. Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

### ExpressJS:

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. It allows to dynamically render HTML Pages based on passing arguments to templates.

MySQL Express Driver:

It is a Node.js module used to interact with MySQL database. To access a MySQL database with Node.js, you need a MySQL driver. Node.js can use this module to manipulate the MySQL database.

Nodemailer:

Nodemailer is a module for Node.js applications to allow sending customised emails.

Fuse.js:

Fuse.js is a powerful, lightweight fuzzy-search library, with zero dependencies. Generally speaking, fuzzy searching (more formally known as approximate string matching) is the technique of finding strings that are approximately equal to a given pattern (rather than exactly). With Fuse.js, you don't need to setup a dedicated backend just to handle search.

## Database:

MySQL:

MySQL is a widely used relational database management system (RDBMS). MySQL is free and open-source. To build a web site that shows data from a database, you will need an RDBMS database program (like MySQL).

## Features of VueJS:

Virtual DOM:

VueJS uses virtual DOM, that is also utilized by many other frameworks like React, Ember, etc. The changes are not made to the DOM, instead a duplicate of the DOM is generated which is existing in the form of JavaScript data structures. Whenever any changes are to be required, they are done to the JavaScript data structures and the replica is compared with the original data structure. To the real DOM, the final changes are then updated which the user will see changing. In terms of optimization, this approach is good and also it is less expensive and the changes required are done at a faster rate.

Data Binding

This feature of data binding assists in assigning or manipulating the values of HTML attributes, change the style, assign classes with the help of binding directive called **v-bind** available with VueJS.

Components

One of the biggest and most important features of VueJS is components. Components assists programmers in creating custom elements, that can be reused in HTML.

## Event Handling

To listen to the events in VueJS, **v-on** is the attribute added to the DOM elements.

## Animation/Transition

VueJS offers different ways to provide transition to HTML elements when they are updated/created or deleted from the DOM. VueJS uses an in-built transition component that requires to be enclosed around the element for transition effect. Third party animation libraries to add more interactivity to the interface can be easily added.

## Computed Properties

One of the most important facets of VueJS is computed properties. It assists users in listening to the changes made to the UI elements and performs the required calculations. No additional coding is required.

## Templates

VueJS offers HTML-based templates that bind the Vue instance data with the DOM. Vue then compiles these templates into virtual DOM Render functions. We can make use of the template of the render functions and to do so we have to replace the template with the render function.

## Directives

VueJS has some in-built directives such as v-bind, v-on, v-model, v-show, v-if and v-else, which can be used to do different actions on the frontend.

## Watchers

Watchers are used on data that changes. For example, form input elements. Here, there is no need of adding any additional events. Watcher supervises handling of any data changes, thereby making the code simple and fast.

## Routing

With the assistance of Vue-Router library, navigation between pages is performed without refreshing the page every time when it is loaded.

## Lightweight

The performance in VueJS is very fast and is also very lightweight.

## Vue-CLI

Users can build and compile their project/application easily by using Vue-CLI. Vue-CLI stands for “Vue-Command Line Interface” and users can install VueJS by using this feature.

## VueJS vs other famous frontend javascript frameworks:

### VueJS v/s React

#### Virtual DOM

A virtual representation of the DOM tree is virtual DOM. A JavaScript object can be created with the help of virtual DOM. This javascript object created using virtual DOM is the same as the one created with a real DOM. Any time the DOM requires a change to be made, a new JavaScript object is created and to it the changes are done. Later, the final changes are updated in the real DOM after comparing both the JavaScript objects.

React and VueJS both utilize virtual DOM, which makes it faster.

#### Template v/s JSX

In VueJS, html, css and js are used separately. Beginners can easily understand and adopt to the VueJS style. The template based technique used in VueJS is very easy.

JSX approach is used in React. In React everything is written in JavaScript including THE HTML and CSS parts.

#### Installation Tools

VueJS uses **vue-cli /CDN/npm** and React uses **create react app**. Both the frameworks are easy to use and the project is built with all the basic files needed. VueJS does not need webpack for the build, whereas react does. We can start with VueJS coding anywhere by using the cdn library.

#### Popularity

VueJS is less popular than react among the web developer community. Also, job opportunity is more in react than it is in VueJS. Facebook uses React, which is the primary reason React is more popular than any other frontend javascript frameworks. React follows the best practice of JavaScript, since its core concept is javascript. Developers who use React are usually those having very good knowledge of javascript concepts.

VueJS is a developing framework that is in continuous ascendance. As mentioned before, the job opportunities available in VueJS are less when compared with React. But according to a recent survey, many people are currently adapting to VueJS, which can lead to VueJS overtaking its other counterparts in terms of popularity in the near future. There is a good community working on the different facets of VueJS. The vue-router library is maintained by this community, who release regular updates.

VueJS has built a powerful library by taking the good parts from other frontend frameworks such as React and Angular. Owing to its lightweight library, VueJS is much faster in comparison to Angular or React framework.

VueJS v/s Angular

**Similarities**

VueJS shows a lot of similarities with Angular. Directives used in VueJS such as v-for, v-if are very similar to ngFor, ngIf used in Angular. Both frameworks use a command line interface for project installation and to build the project. Vue-cli is used in VueJS and angular-cli is used in Angular. Both provide server-side rendering, two way data binding, etc.

**Complexity**

Beginners can easily understand and adopt to the VueJS style. As mentioned earlier, a beginner can use the CDN library of VueJS and get started in any javascript compiler. Whereas for Angular, a user needs to follow through a number of steps to install it. So, Angular is difficult for beginners to get started with web development. Developers coming from pure javascript background find Angular difficult to use since coding is done using TypeScript. However, learning Angular is easy for users coming from Java and C# background.

**Performance**

It is up to the users to decide the performance. But, the file size of a VueJS project is much lighter than in comparison with Angular.

**Popularity**

Currently, VueJS is less popular than Angular. Many organizations prefer using Angular, making it a very popular choice for frontend development. Job opportunities are also high in Angular. However, due to VueJS's continuous ascendance among the frontend developer community, it can be considered as a good competitor for Angular and React.

**Dependencies**

Angular offers many in-built features. We just have to import the modules required and get started with it, for example, @angular/form, @angular/animations.

VueJS does not match Angular in terms of having built-in features. It depends on a third party libraries to work on it.

**Flexibility**

Without any problems, VueJS can be easily integrated into big projects for front-end development. Angular is not that good in terms of its flexibility.

**Backward Compatibility**

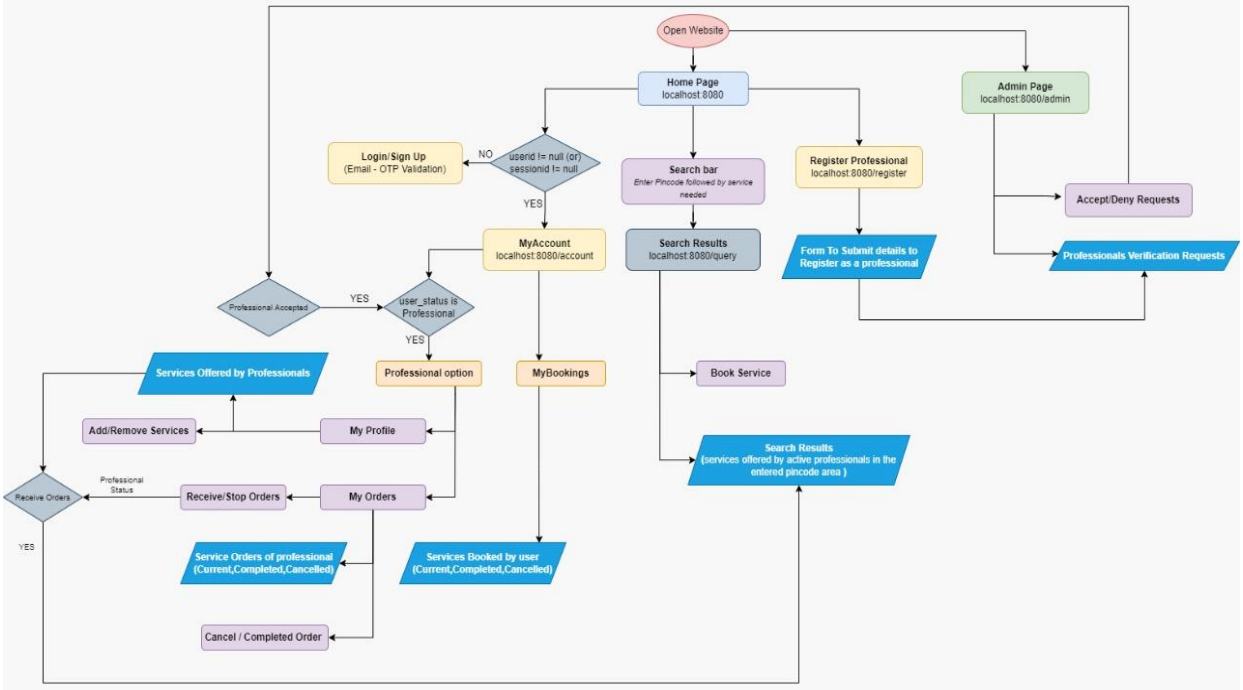
Initially, there was AngularJS, Angular2 and now Angular4 is available. Angular2 and AngularJS display vast difference. Because of the core differences between Angular JS and Angular2, project application that was developed in one cannot be converted to the other.

The recent version of VueJS available is 3.0 and it shows very good backward compatibility. It offers good documentation, which makes it easier to understand.

**Typescript**

Coding is done using TypeScript in Angular. To get started with Angular, users are required to have prior knowledge of Typescript. We can start with VueJS coding anywhere by using the cdn library. We are able to work with standard JavaScript, which is very easy to start with.

## FLOWCHART OF OUR PROJECT:



## IMPLEMENTATION SCREENSHOTS:

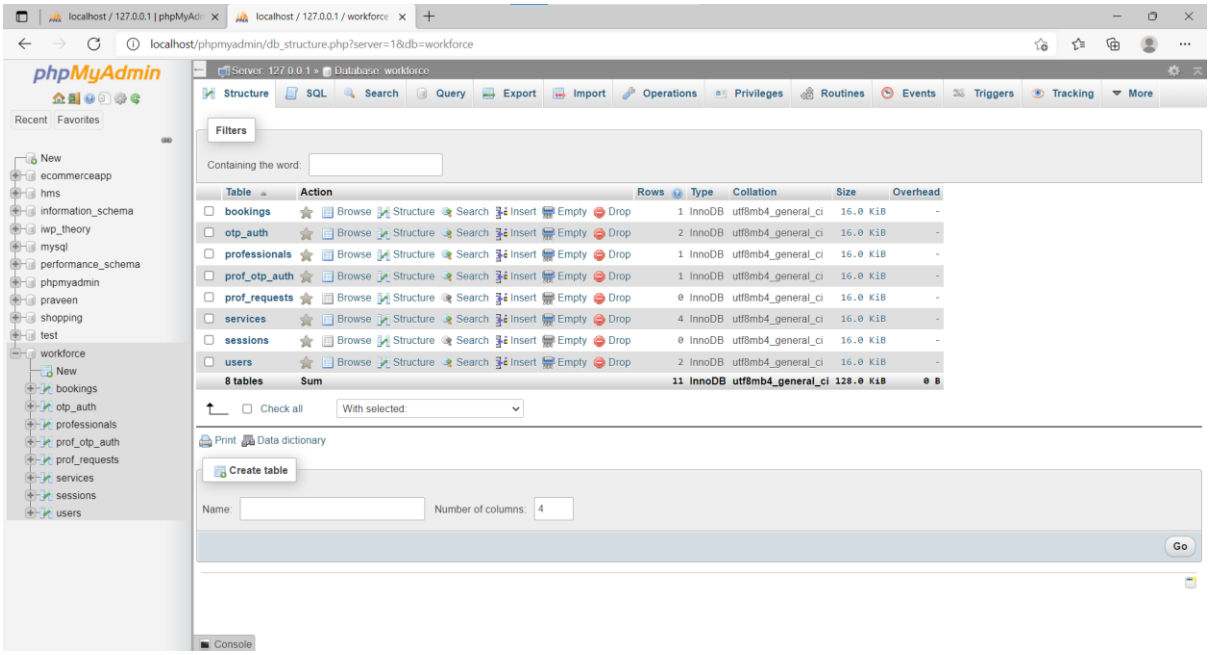
WorkForce logo:

WorkForce

Run the XAMPP server and open the “WorkForce” database using the MySQL module in the XAMPP SERVER.

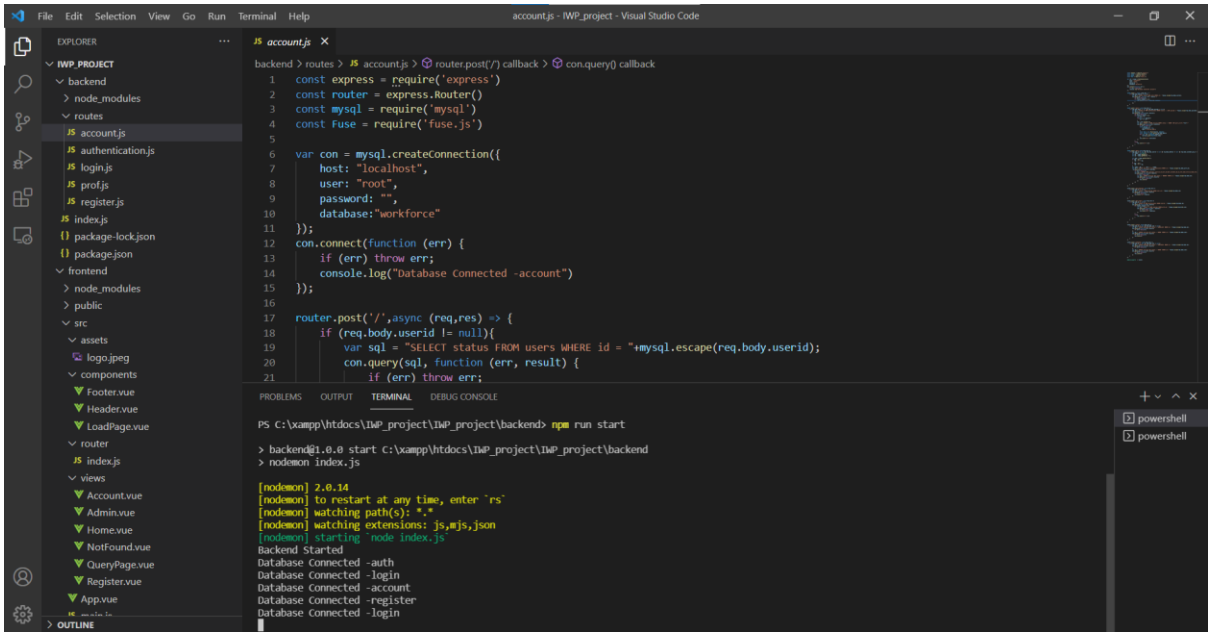
WorkForce database with 6 tables:

- Bookings
- Otp_auth
- Professionals (registered professional’s details)
- Prof_otp_auth
- Prof_requests (requests displayed in admin page of users wanting to become a service provider)
- Services (Details of the services offered by professionals registered in our website)
- Sessions
- Users (Customer details)

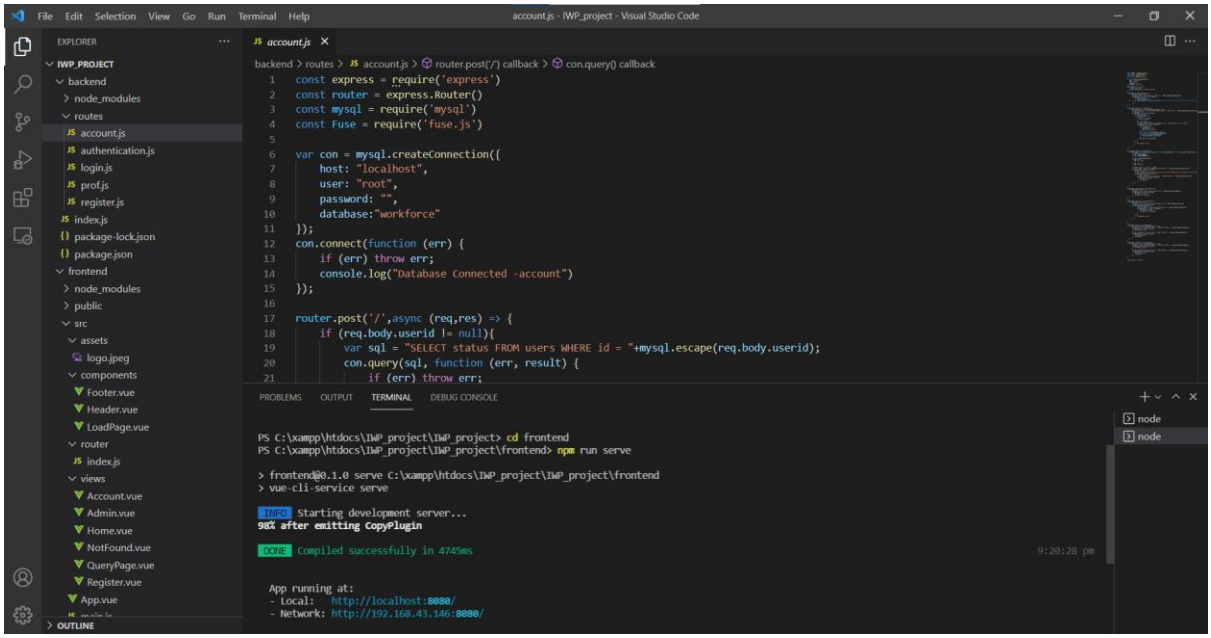



Now open visual studio code and run backend and frontend modules.

Start the backend modules using the command “npm run start”

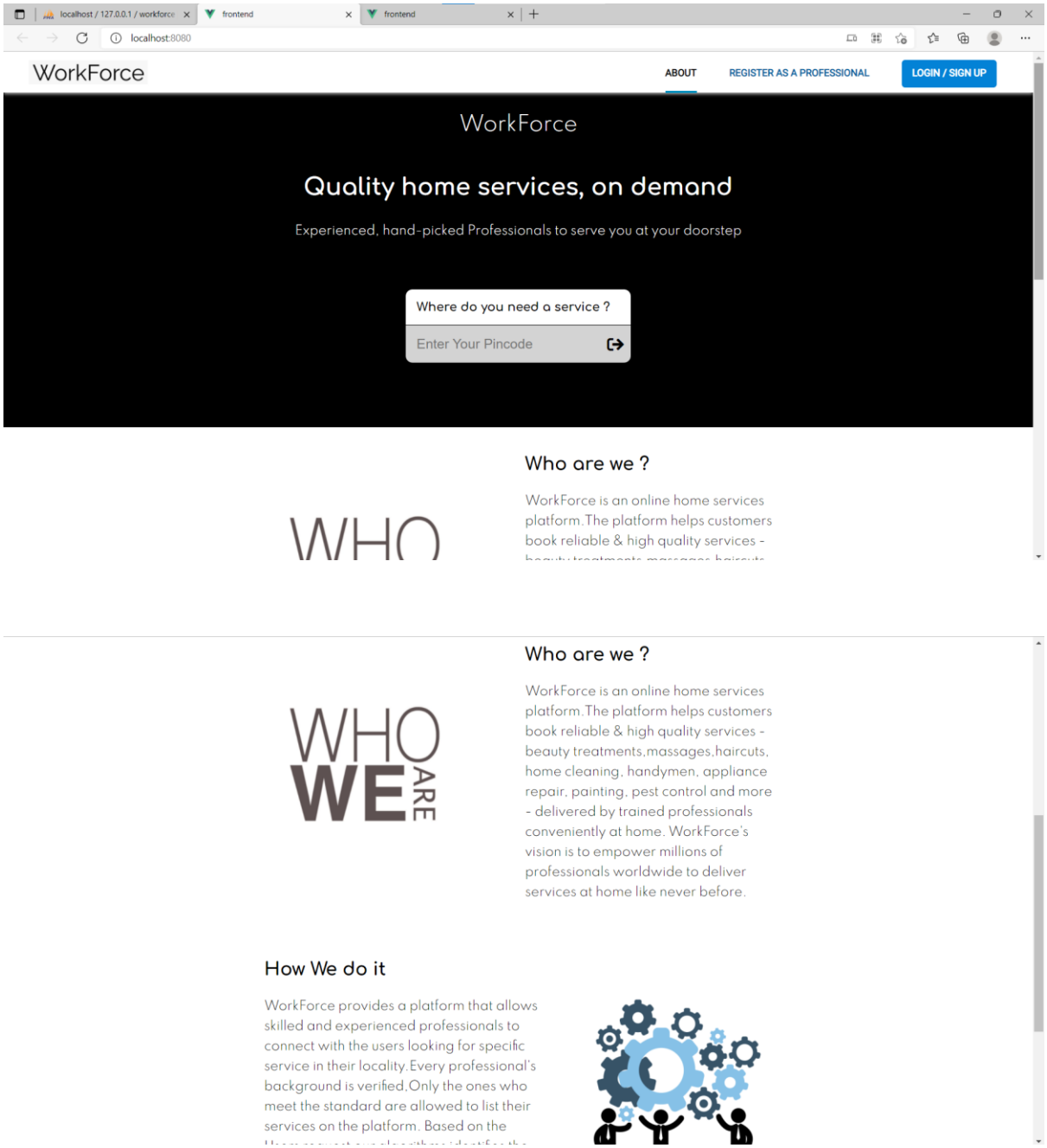



Start the frontend modules using the command “npm run serve”

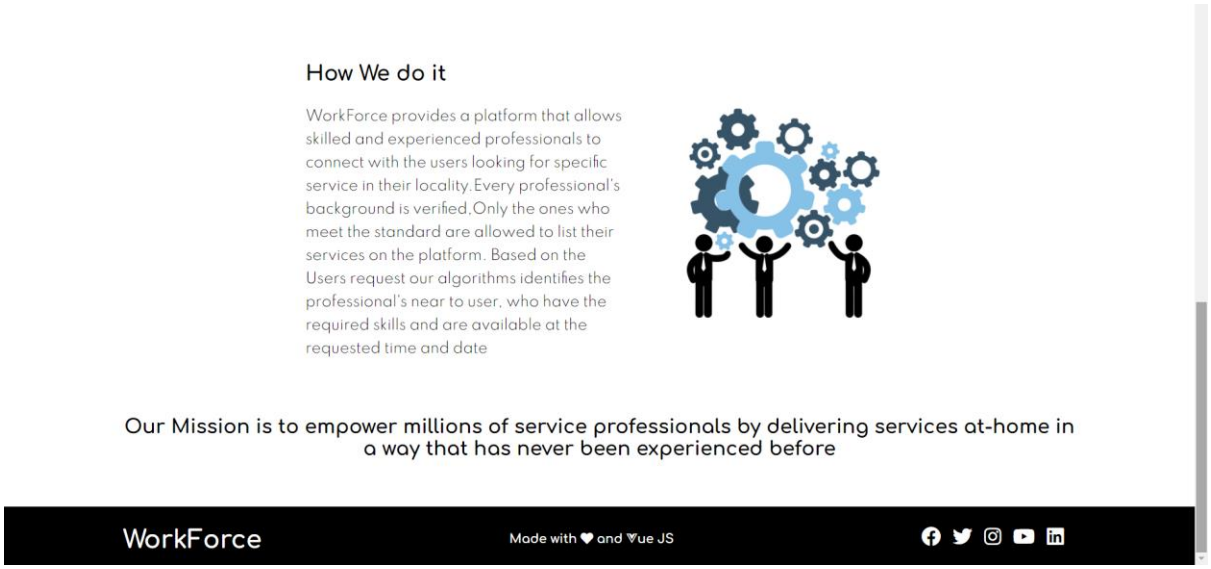



Screenshots of WorkForce website

Home Page

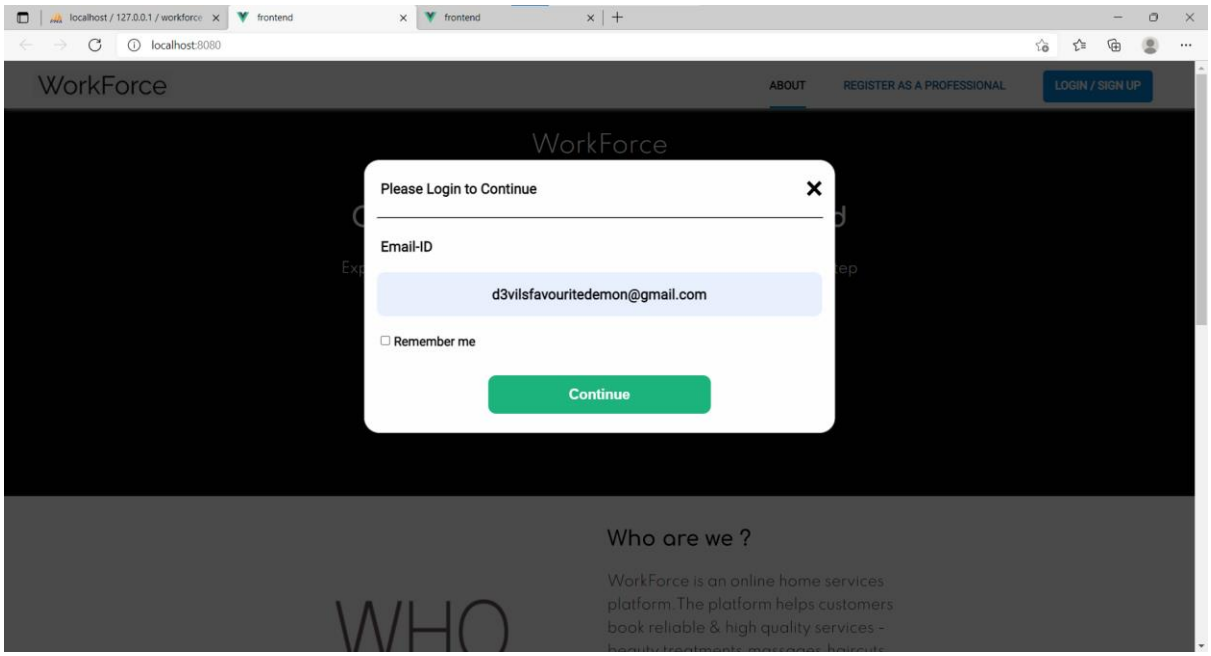

localhost / 127.0.0.1 / workforce
frontend
frontend
localhost:8080
WorkForce
ABOUT
REGISTER AS A PROFESSIONAL
LOGIN / SIGN UP
WorkForce
Quality home services, on demand
Experienced, hand-picked Professionals to serve you at your doorstep
Where do you need a service ?
Enter Your Pincode
WHO
Who are we ?
WorkForce is an online home services platform.The platform helps customers book reliable & high quality services -


WHO
WE ARE
Who are we ?
WorkForce is an online home services platform.The platform helps customers book reliable & high quality services - beauty treatments,massages,haircuts, home cleaning, handymen, appliance repair, painting, pest control and more - delivered by trained professionals conveniently at home. WorkForce's vision is to empower millions of professionals worldwide to deliver services at home like never before.
How We do it
WorkForce provides a platform that allows skilled and experienced professionals to connect with the users looking for specific service in their locality.Every professional's background is verified,Only the ones who meet the standard are allowed to list their services on the platform. Based on the

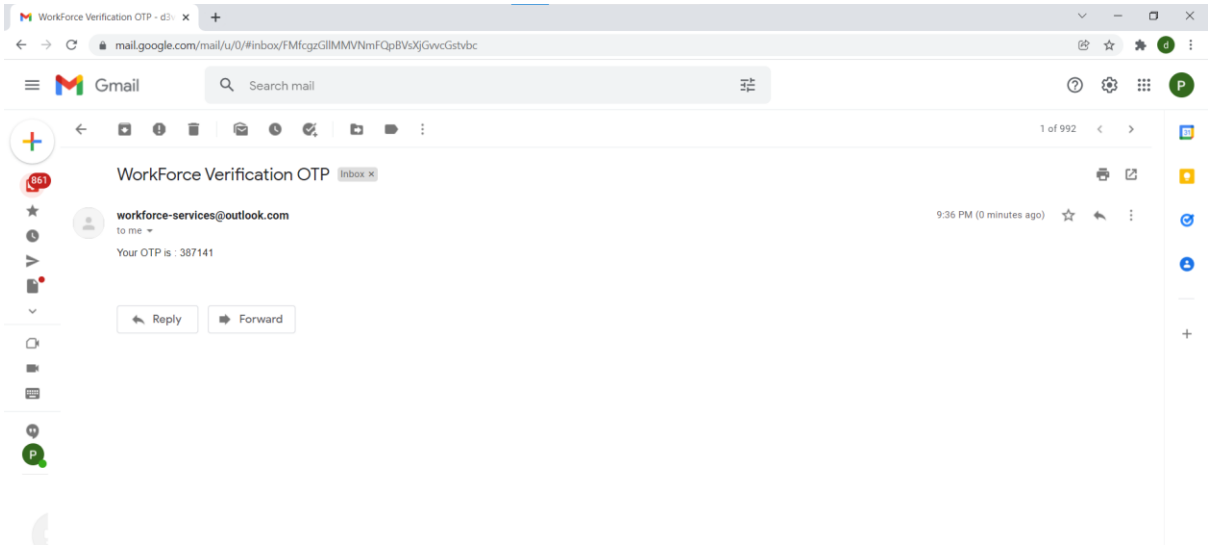



To register as new user, the user needs to click the “Login/Sign Up” button present in the top right corner of our home page

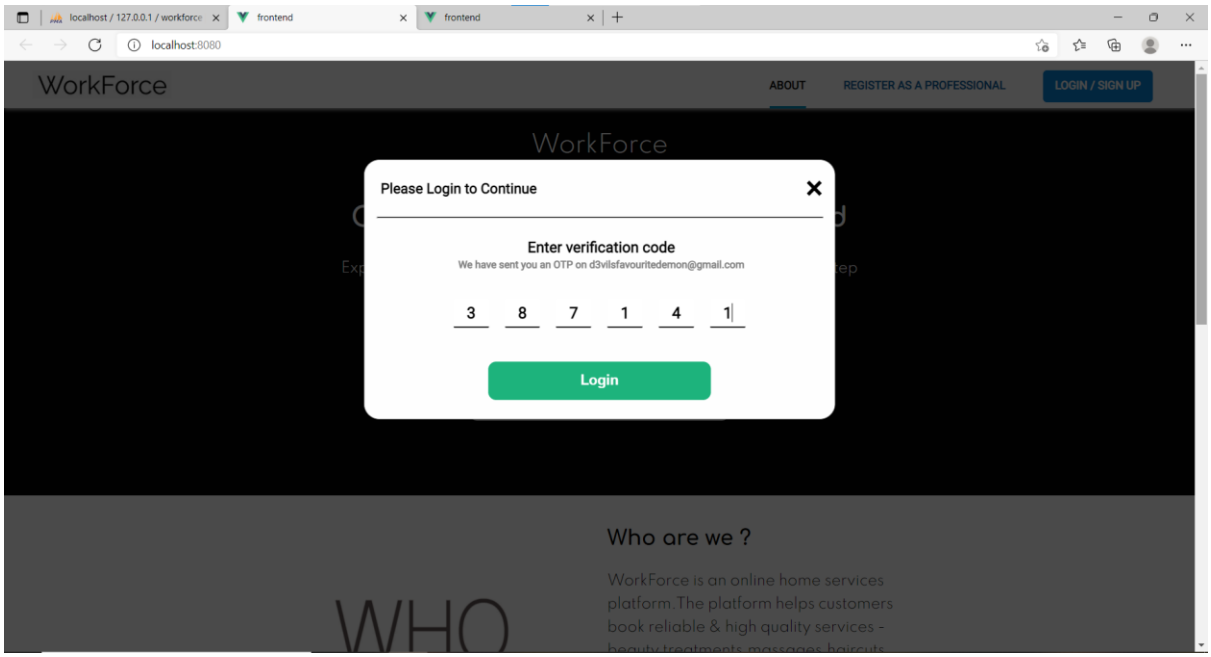
Enter your email ID in the sign-up page and click on the “Continue” that can be seen in the screenshot given below



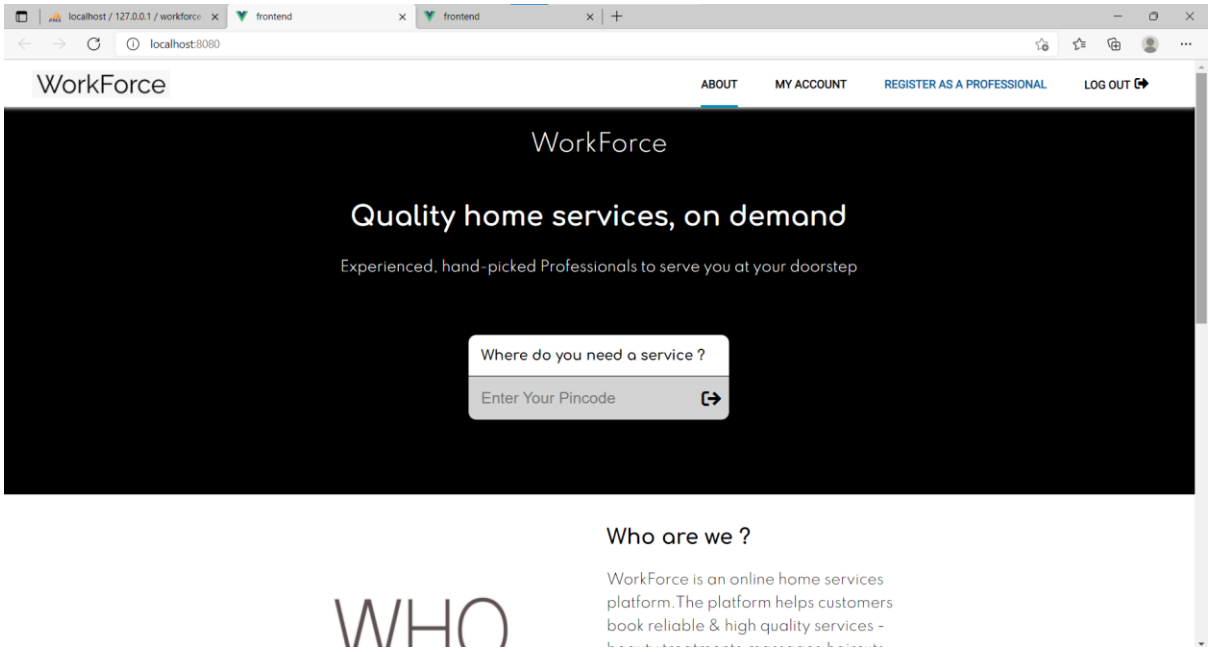

Now, an OTP is sent to the email id mentioned by the user in the sign-up form



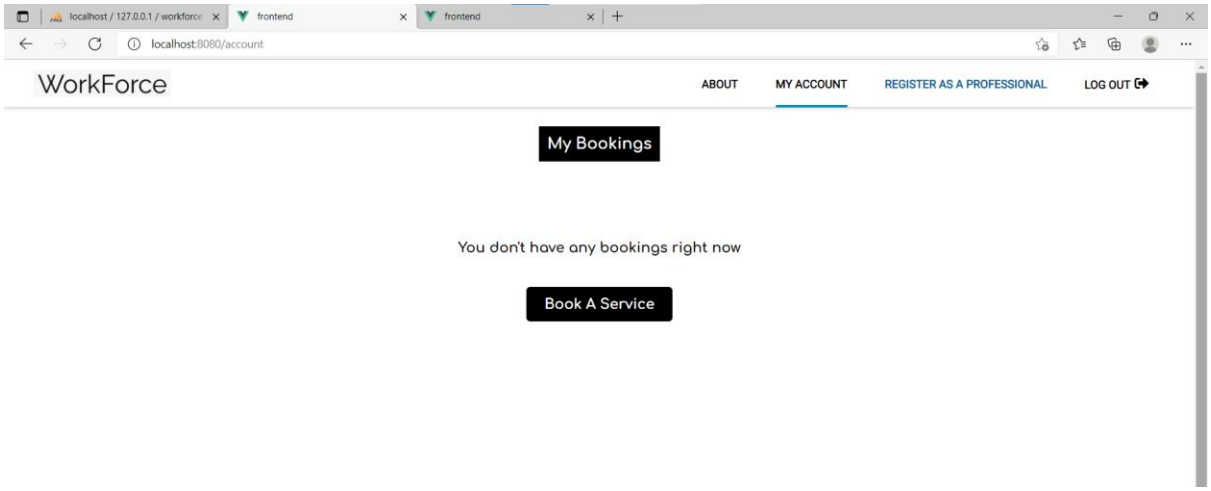

Now, the user needs to enter the otp he received in our website




On entering the correct OTP, the user is now registered in our website

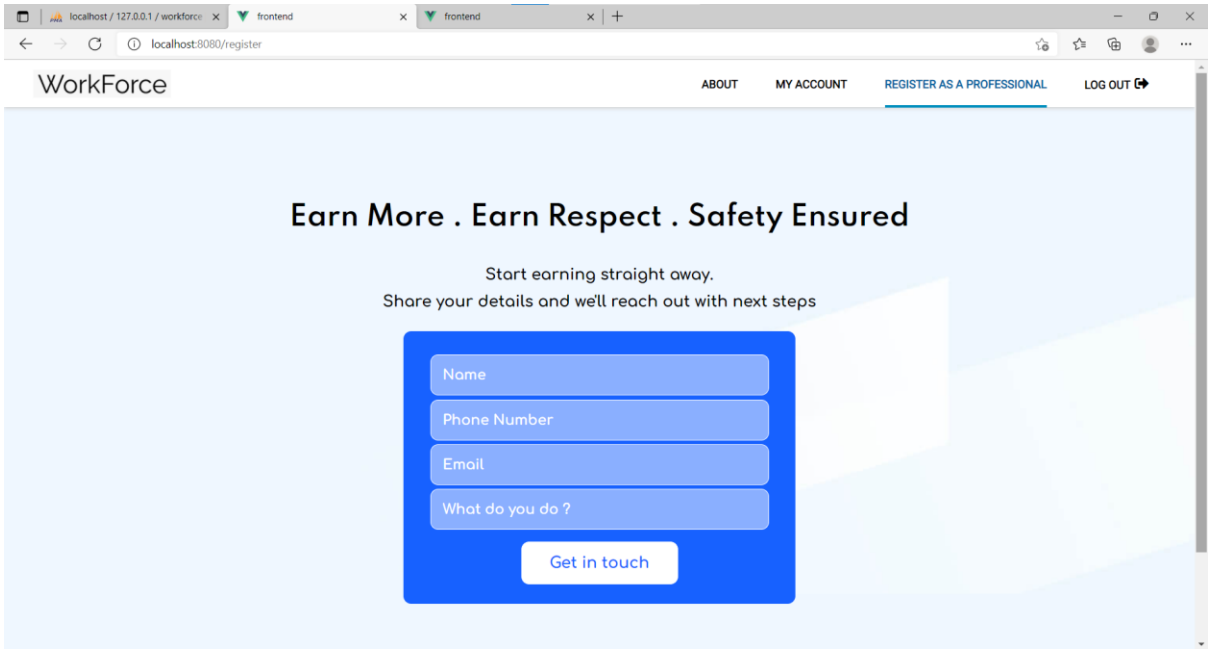


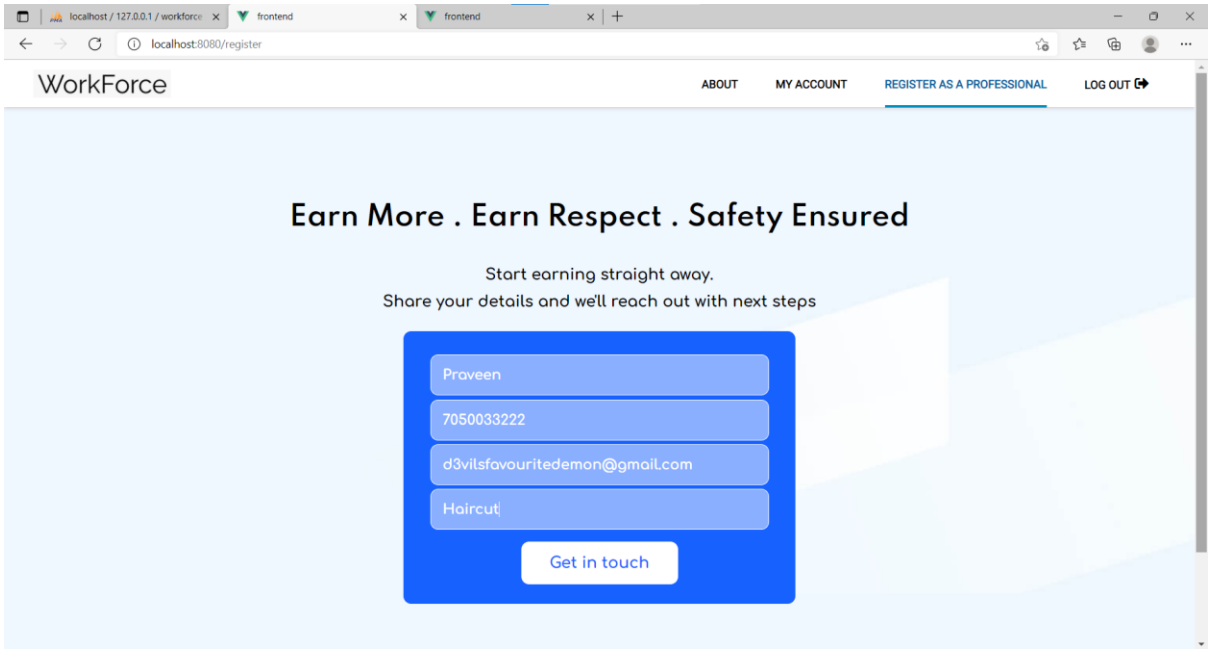

In “My Account” module, the user can book a service and also see the services he has already booked. Since, we have just signed in as a new user there is nothing displayed under My Bookings.



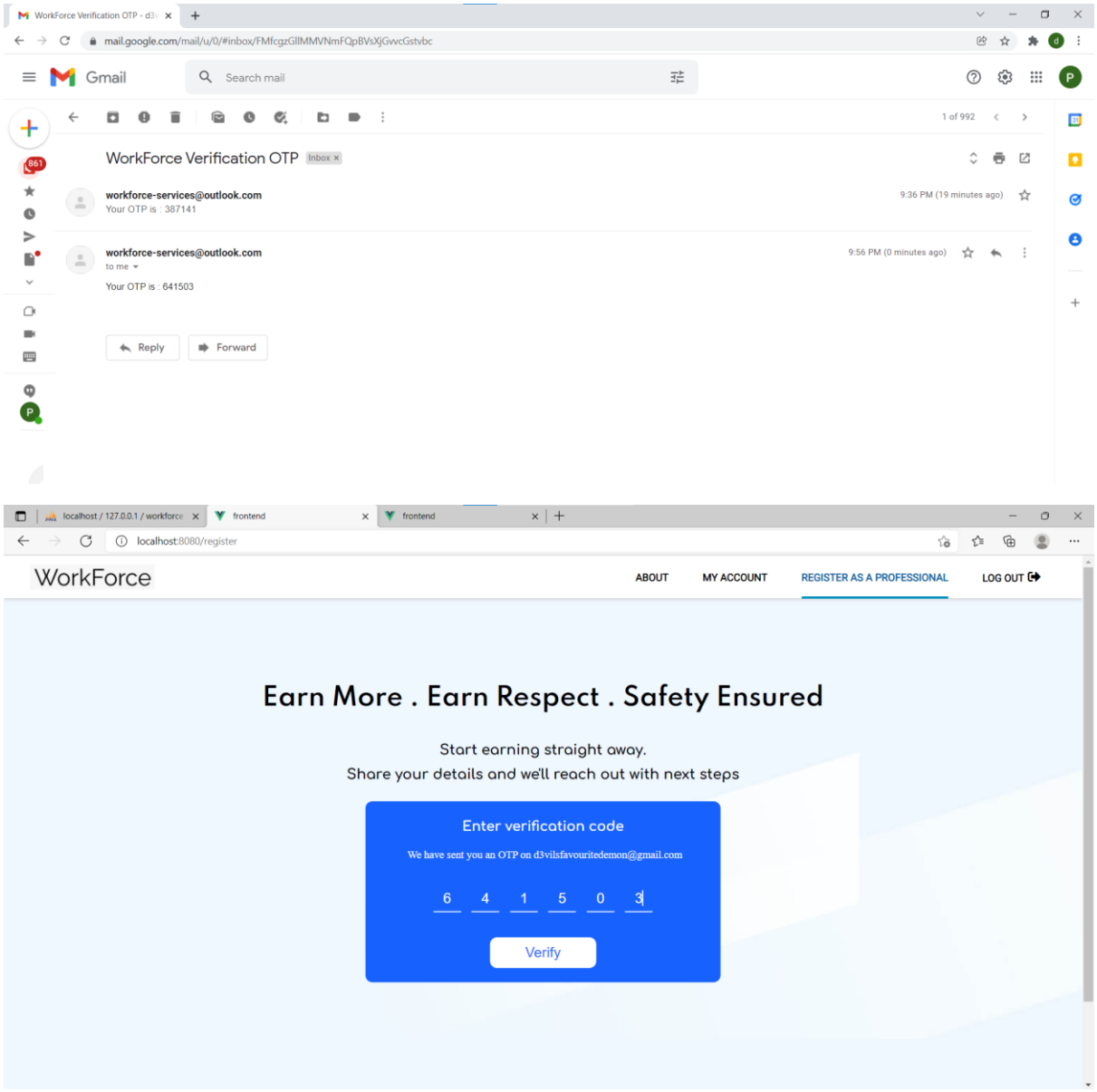

As we can see in the screenshot above, there is an option to “Register As A Professional” in our users’ display. Any user of our website can register as a professional. But whether they are accepted as a service provider depends on our admin.

Register as a professional page:

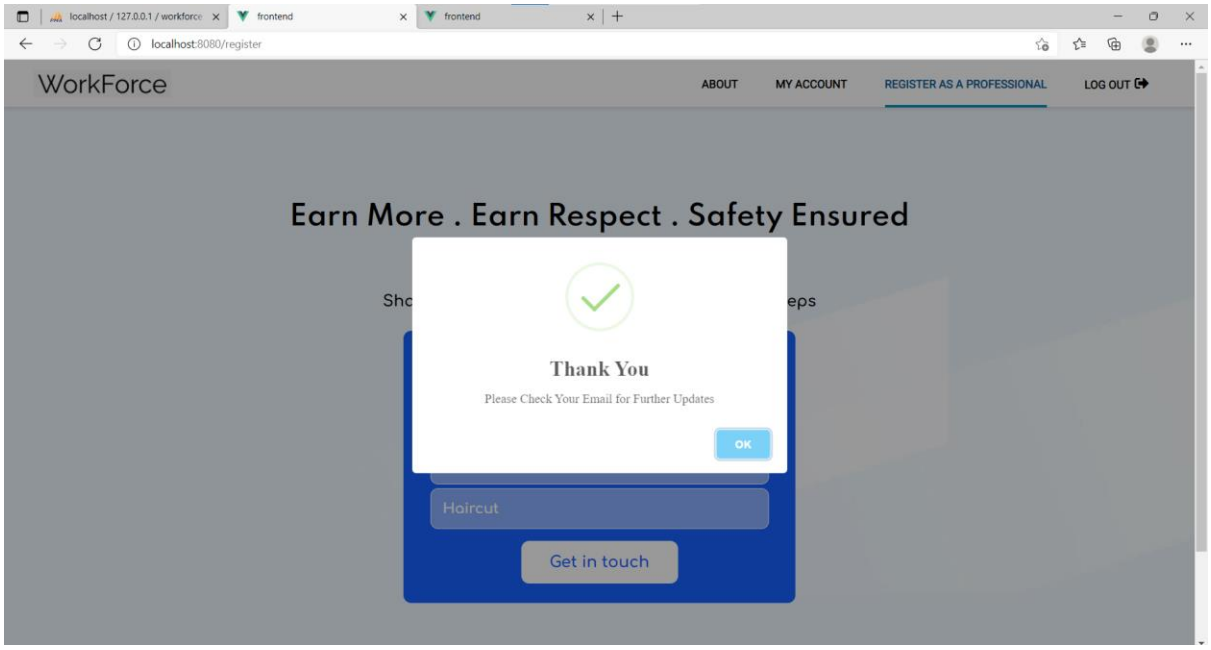



After clicking on the “Get in touch” button, an OTP is sent to the email ID provided by the user while registering as a service provider. Enter the OTP received

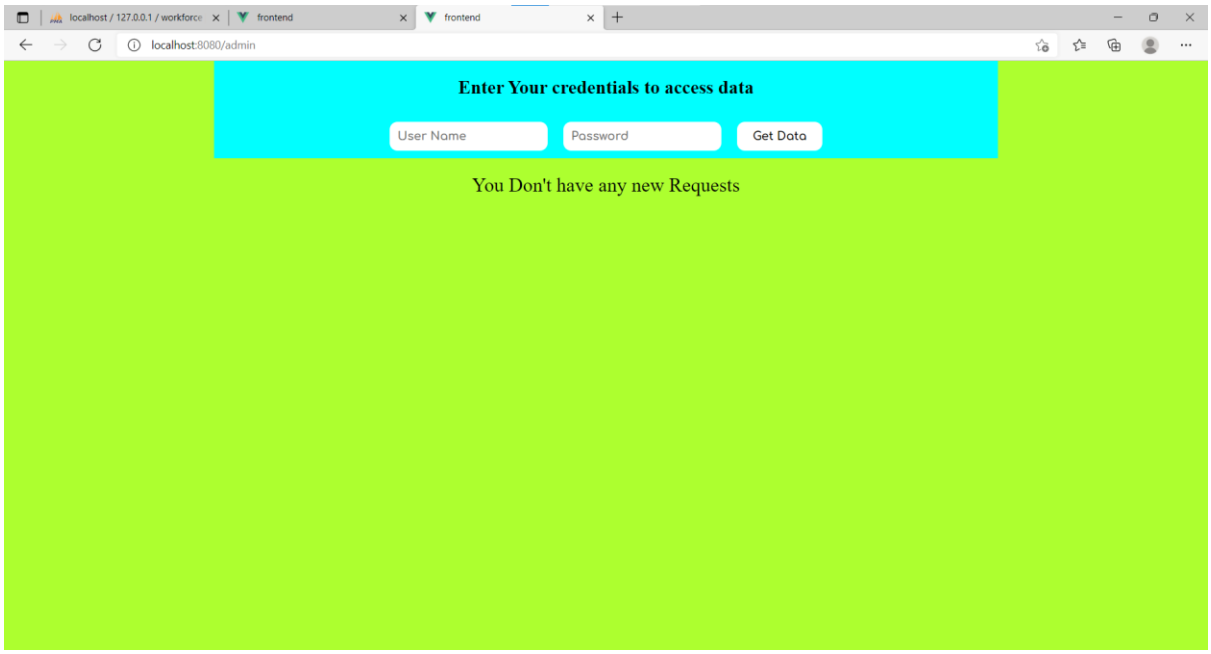



Alert message displayed when the entered OTP is correct



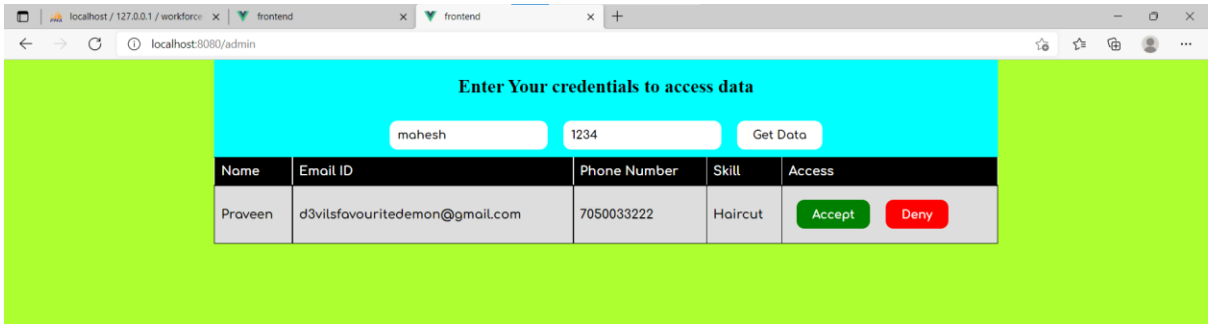

Admin Page:

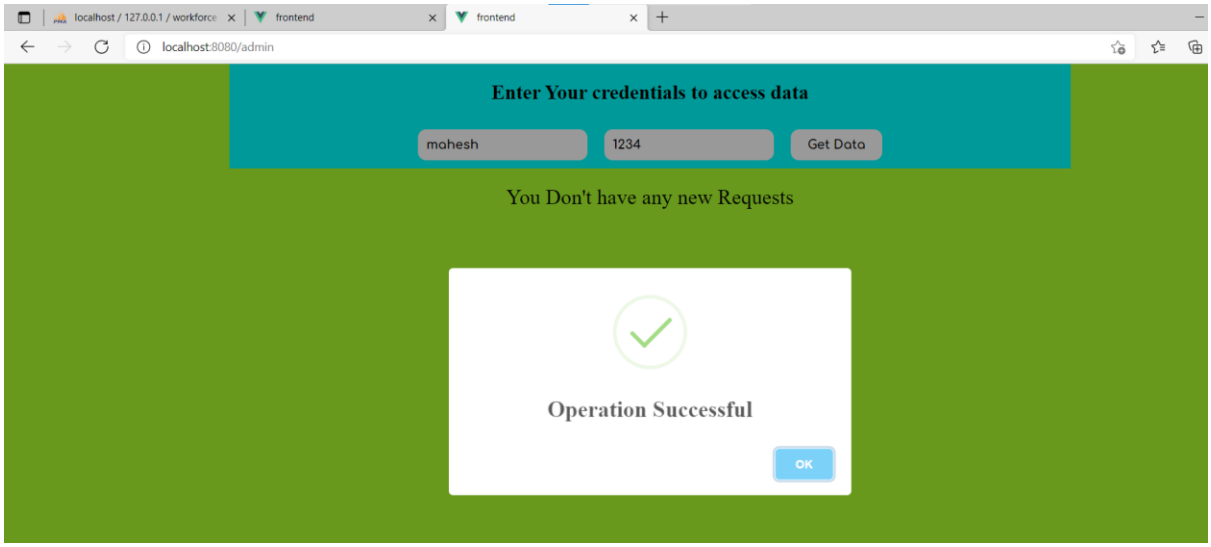
Our admins have the power to accept/deny any requests from the users who have applied to become a service provider. The accepting/denying the request is done based on certain criteria.



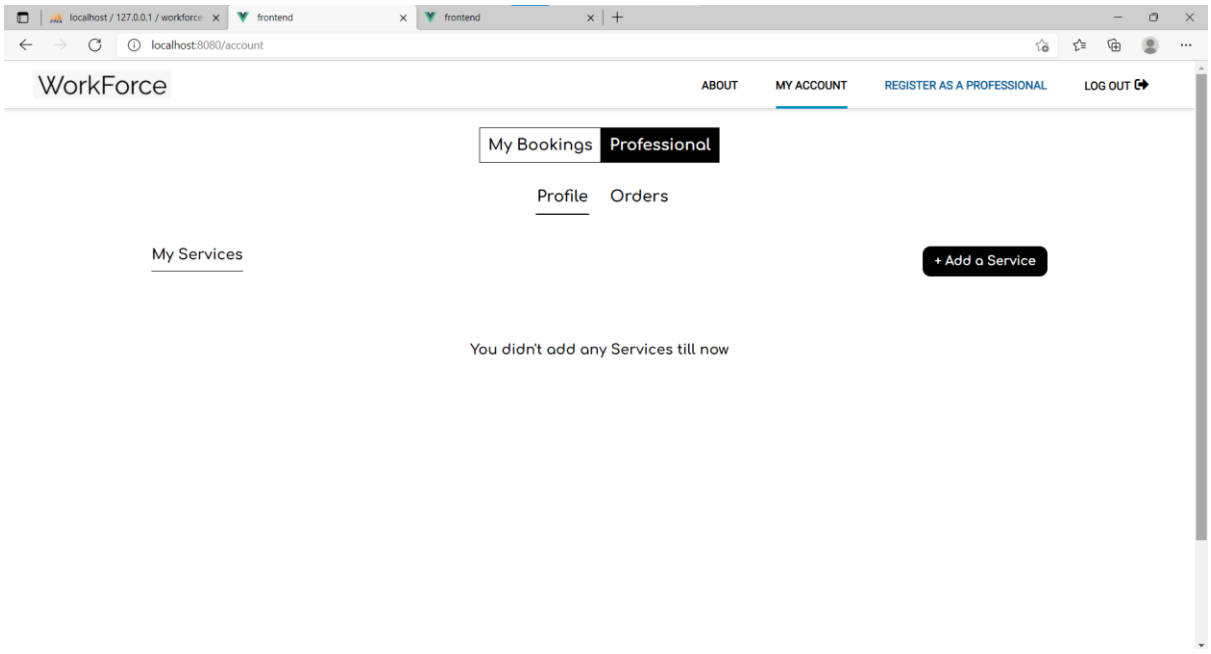

Enter login credentials of the admin to see the requests sent by users who want to become a service provider.

Request sent by the user named Praveen, to become a service provider. Now let us accept the request.



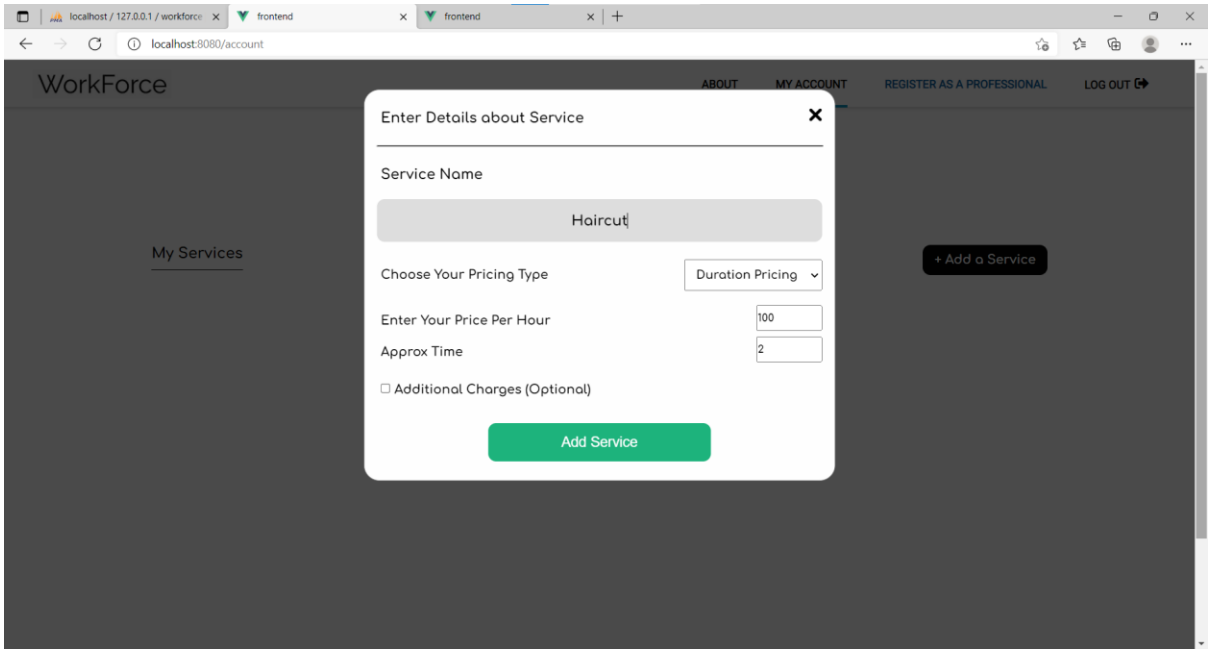

Since, the request is accepted by our admin, the user is now a registered professional under our website. Now, in the My Account page the user can see a “Professional” column in addition to “My Bookings” column.



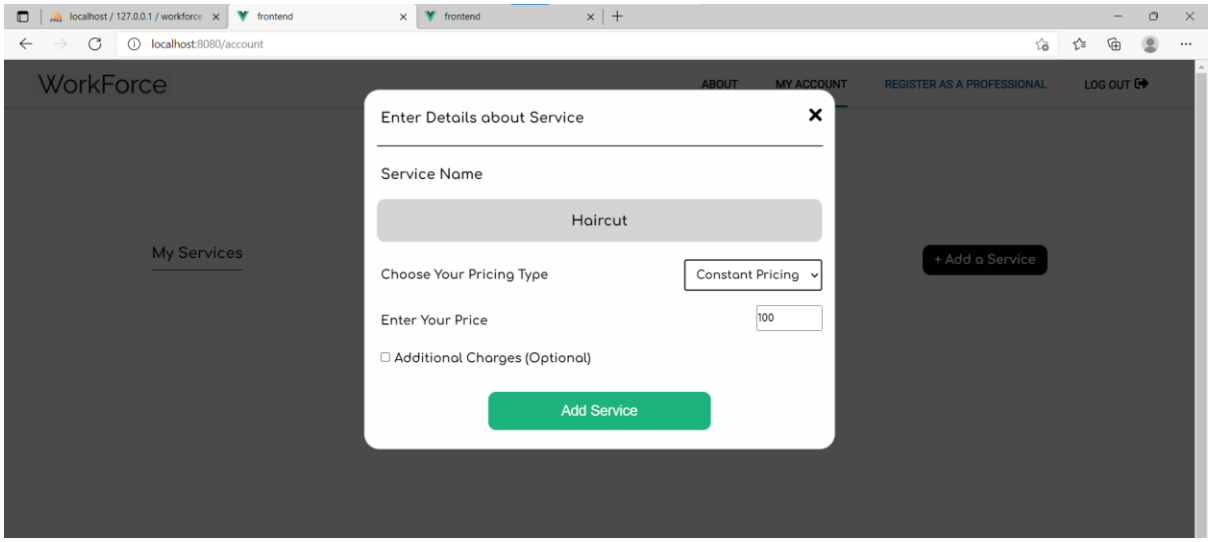
Now, the user can customise his professional page by adding the services offered by him.

On clicking the “Add a service” button, the user will get a form in which he needs to enter the details of the service provided by him. The service provider can choose the pricing type. Pricing type can be either fixed or duration dependent service.

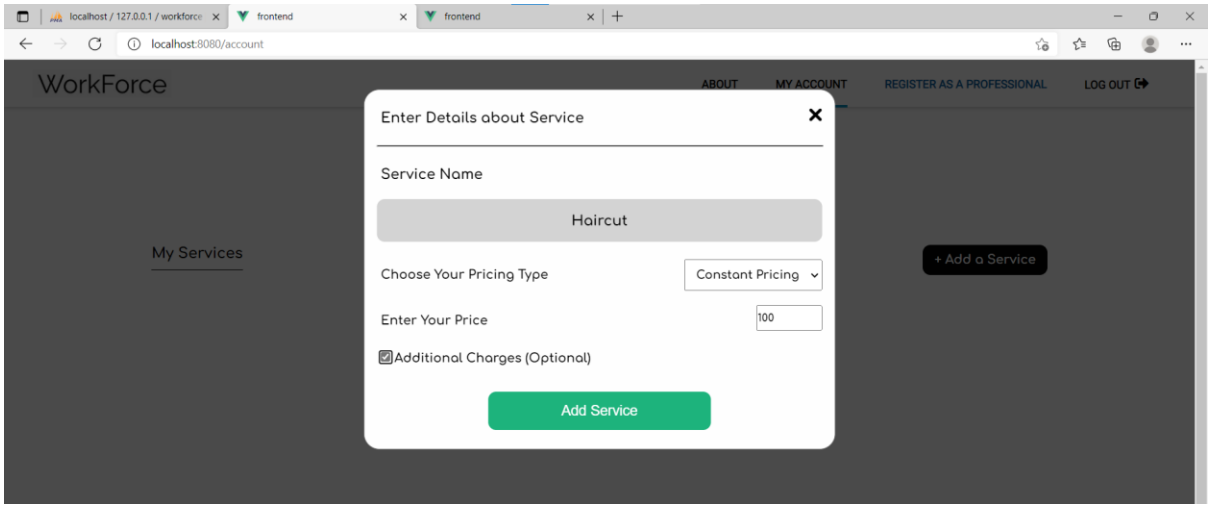
On choosing “Duration Pricing”, the service provider should enter the price he charges per hour and the approximate time for him to do the service.




When the service provider chooses “Constant Pricing” as his pricing type, then he just needs to enter the price he charges regardless of the duration it takes to complete the service.

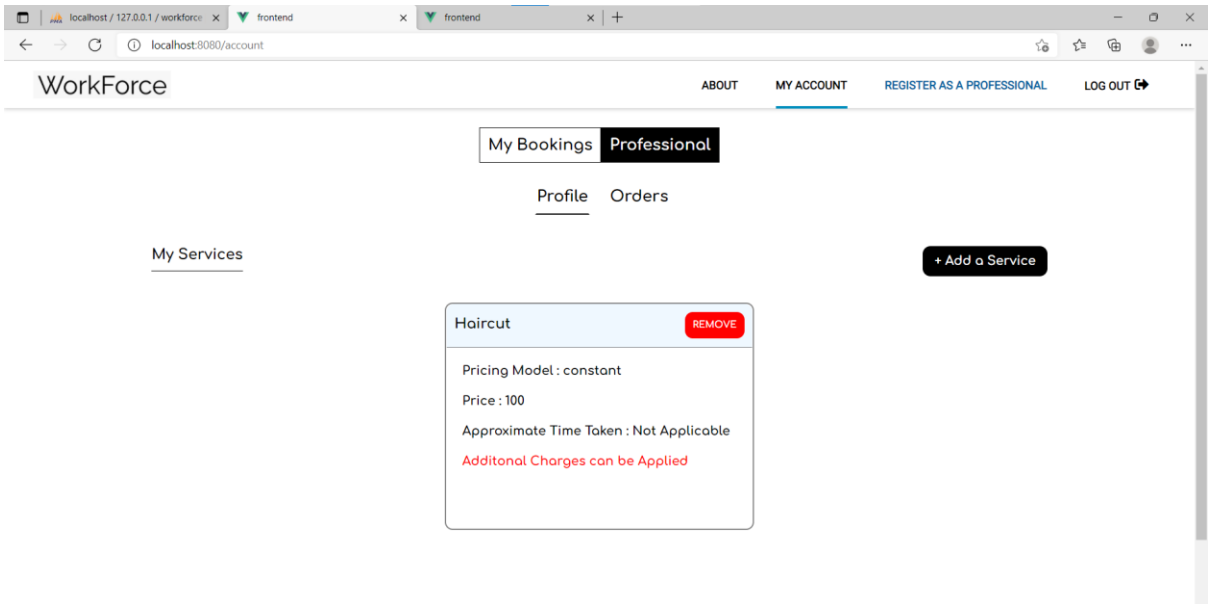



Also, the service provider should check the “Additional Charges” displayed in the form should the service involve buying any materials for the customers in order to do the job.

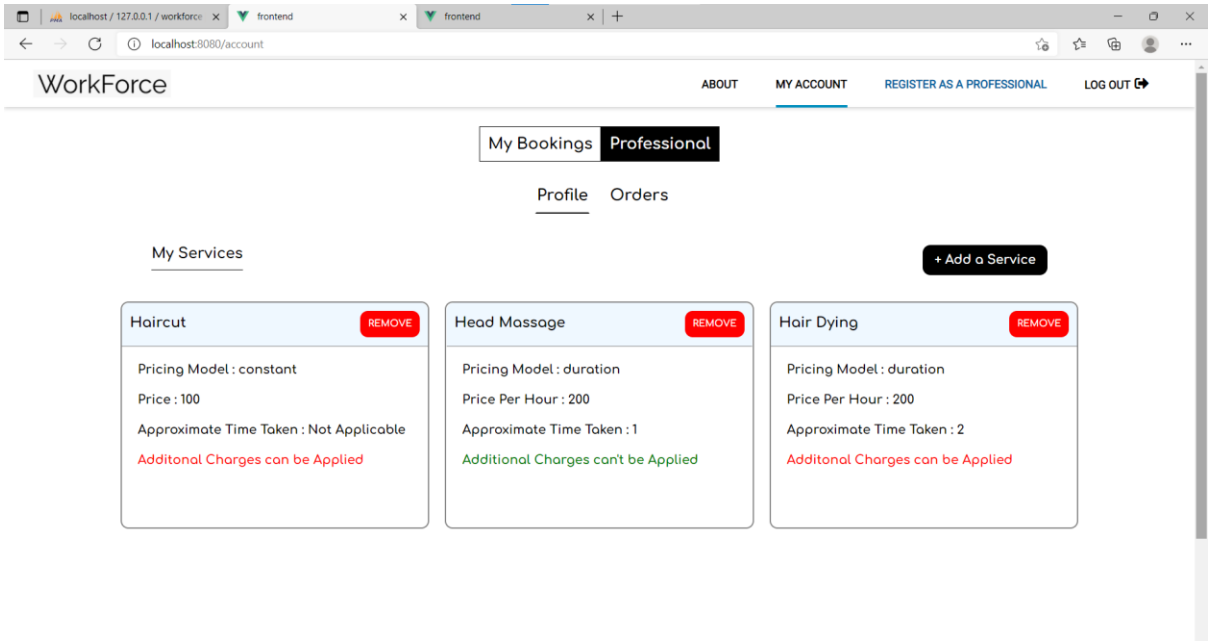



Now, when the “Add Service” button is clicked the service can be seen in the service provider’s “profile” under “My Services”

My Services column is updated.

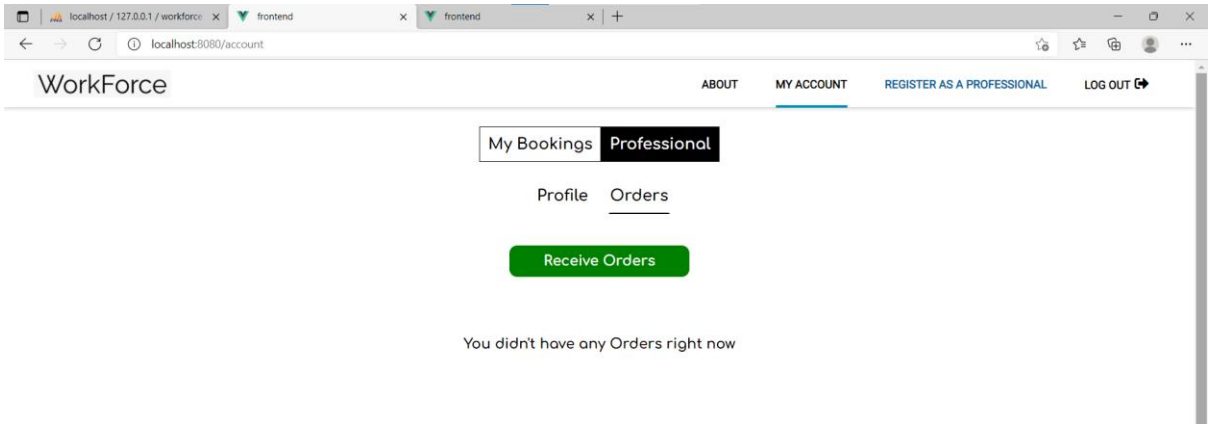



Now, let us add two more services under the same user. The updated profile of the user:



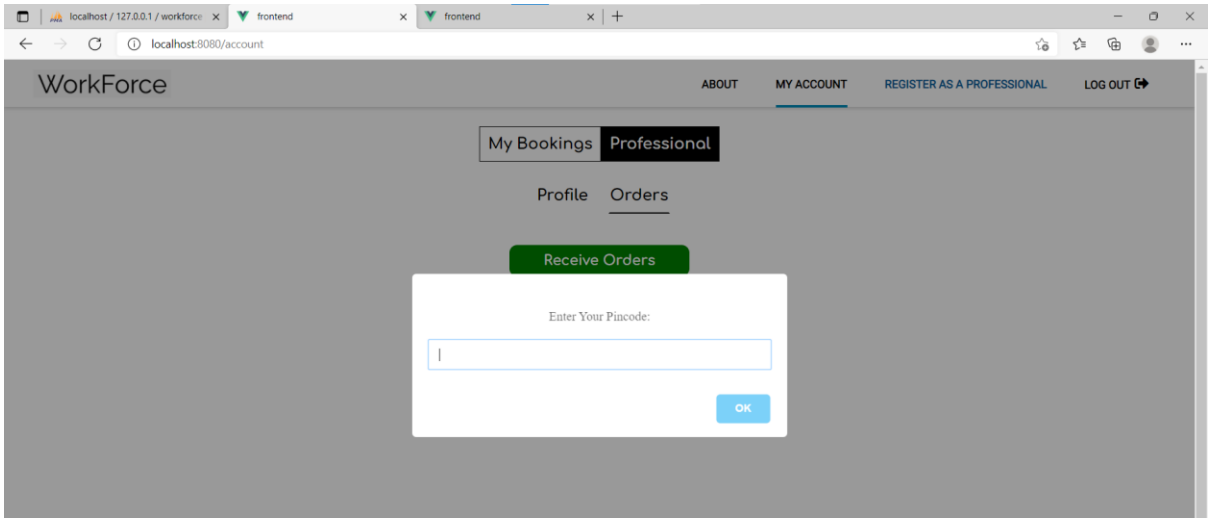

For the registered professional to receive any order, he needs to visit the orders page.

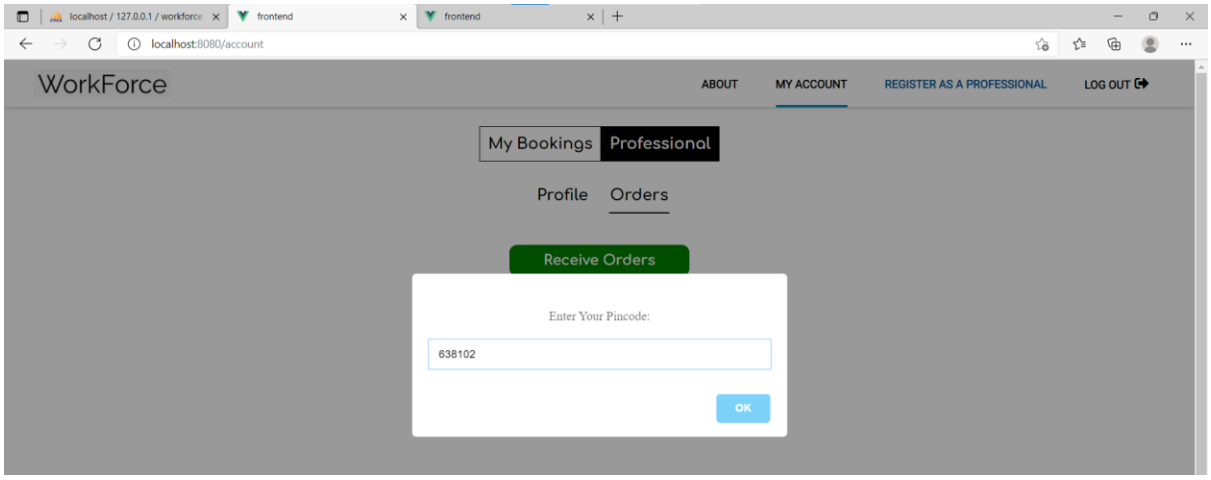
Orders page:



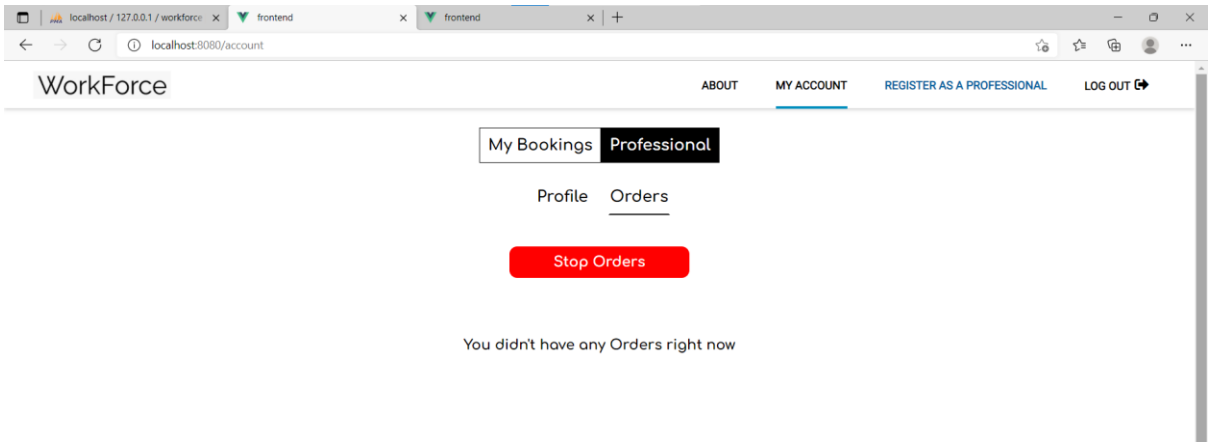

The registered professional should click the “Receive Orders” button to receive any orders.

When the professional clicks the “Receive Orders” button, he is asked to enter the pincode of his current location (where he offers his services)



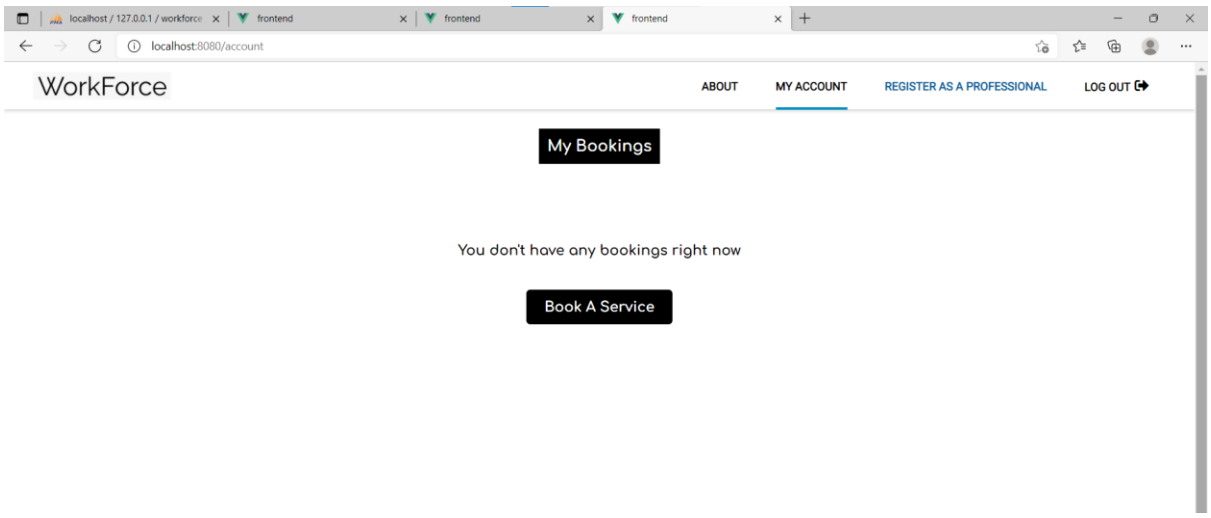




NOTE: The professional receives an order only when he is online. If he goes offline, he won't be listed under the registered professionals, even when the user is searching for the particular service provided by that professional. Nor he receives any order when he clicks the "Stop Orders" button. This ensures that the services are provided on-demand by our professionals.

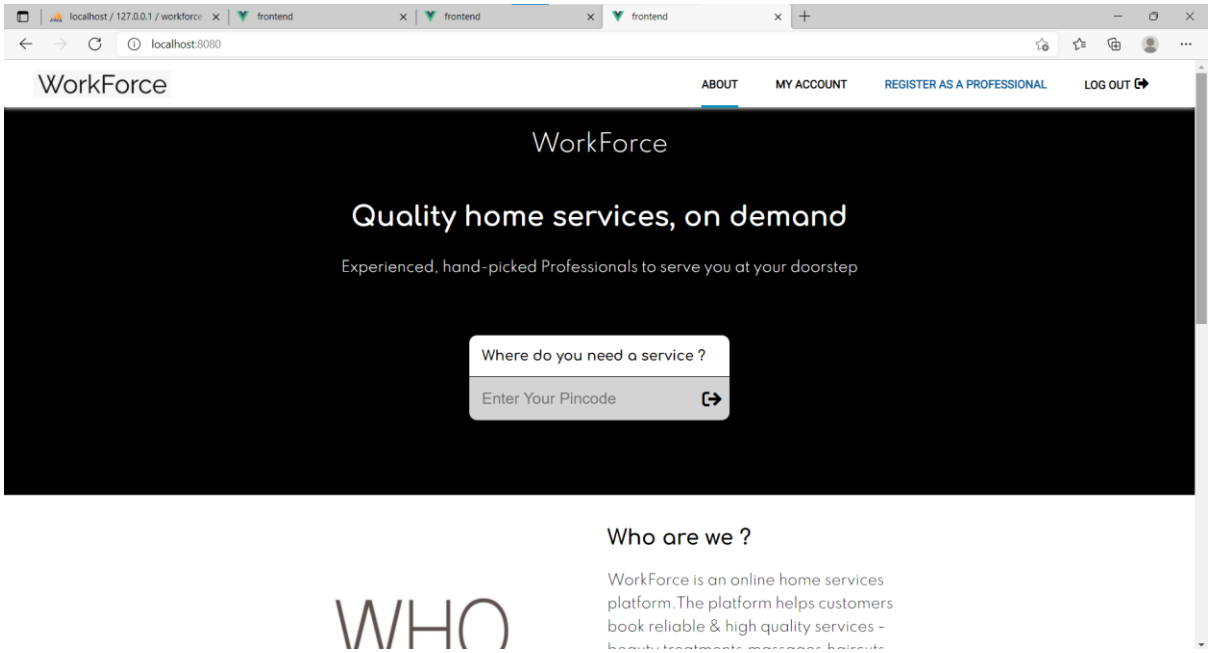



Now, let us try to book a service as a user. To do that let us create a new user.

We are logged in as a new user. Now click on the “Book A Service” button if you wish to book a service.

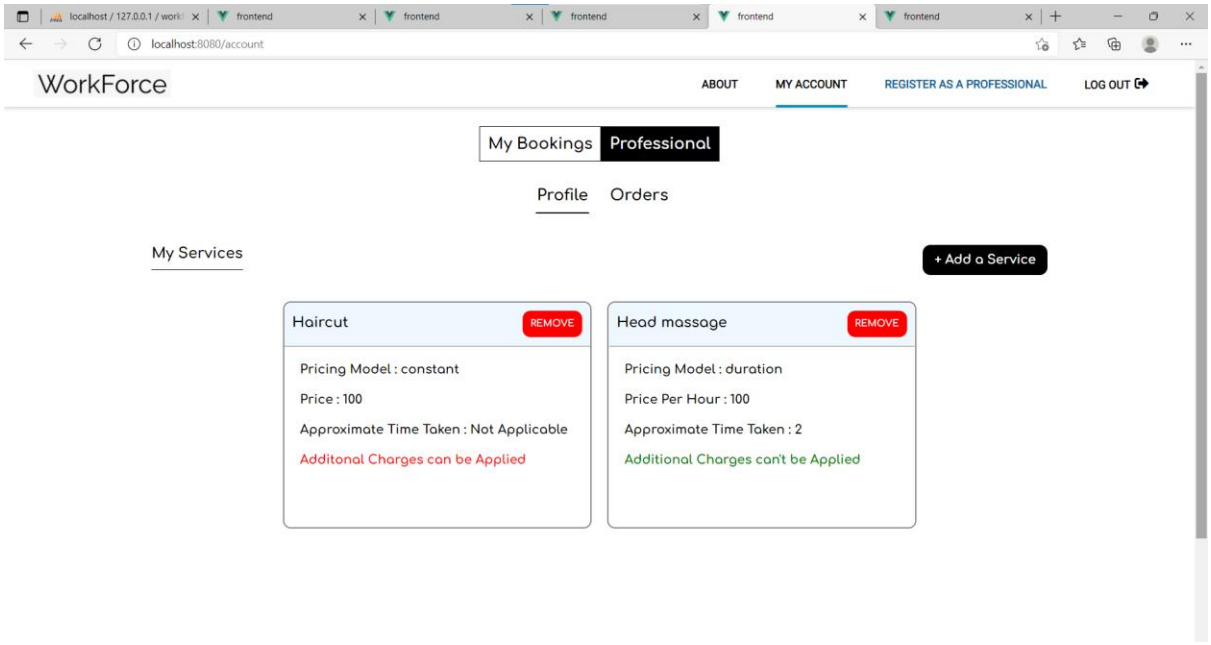


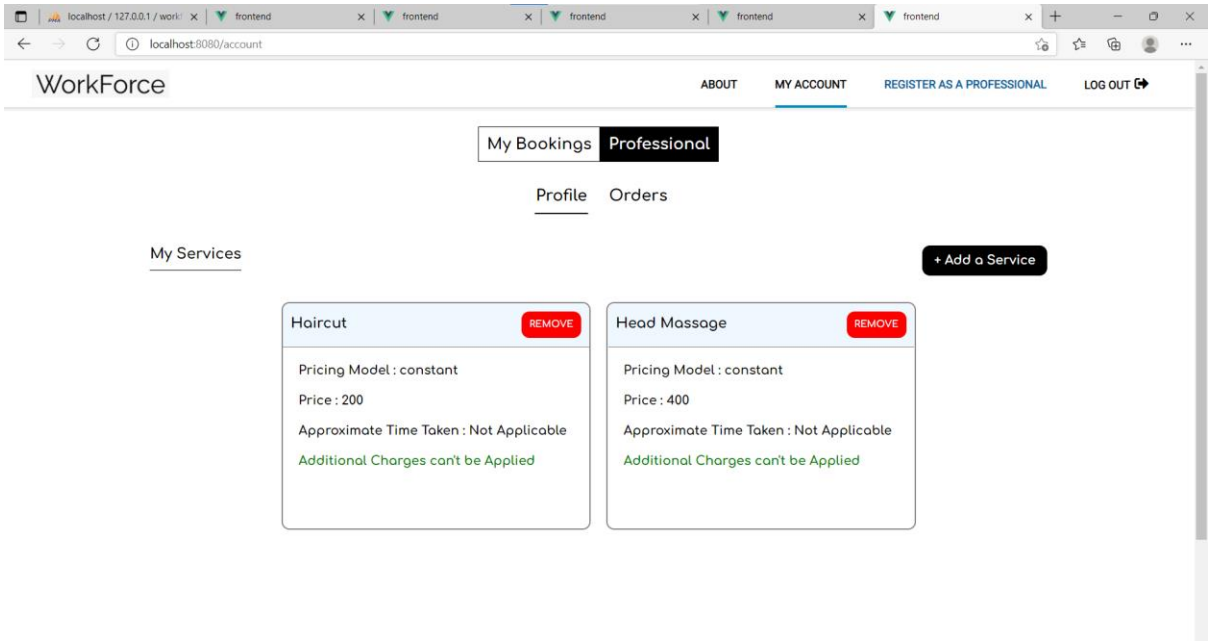

The user is now redirected to the home page where he is asked to enter his pincode. This ensures that the website will only list those professionals who are nearby, for the users to choose.



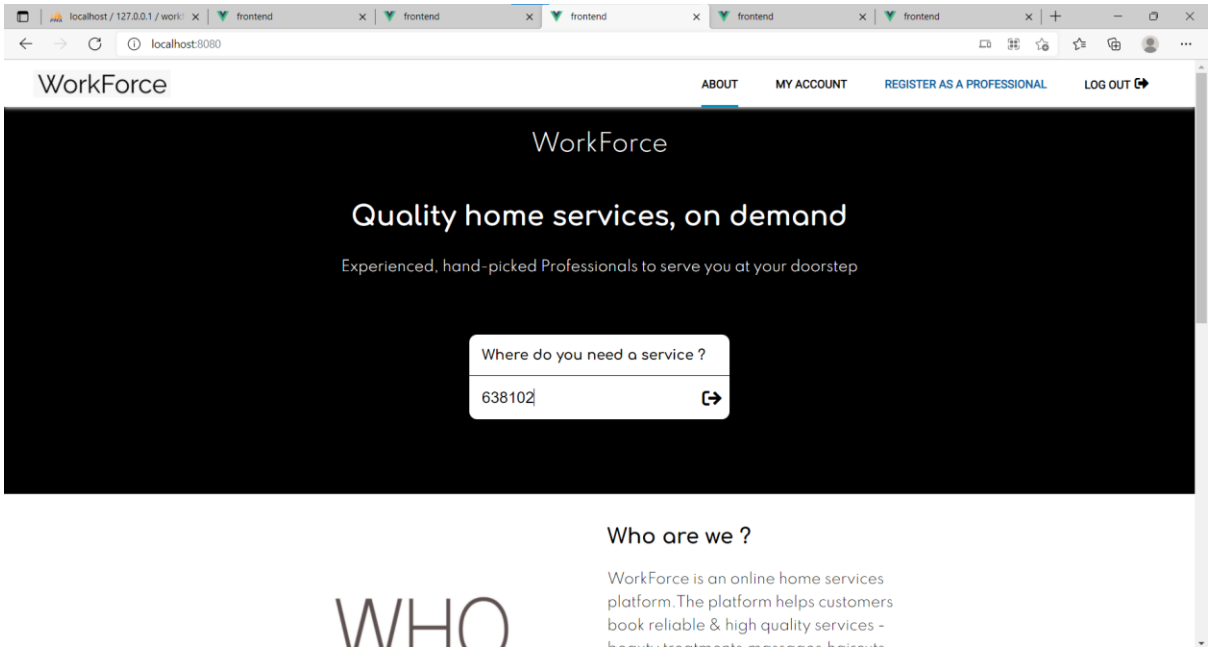

Before searching for the service from the new user account, let us add two more professionals in our website and make them enable the Receive Orders option with pincode: “638102”

Services offered by two newly registered professionals.

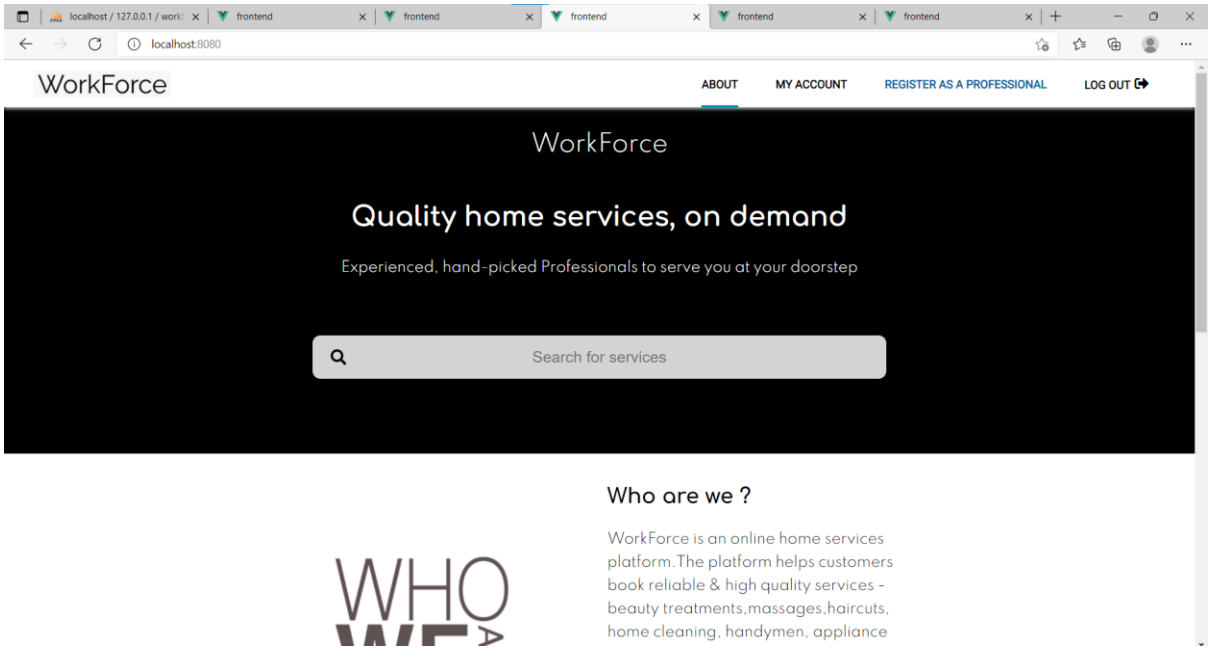



Coming back to the users' page. The user who wants to book a service first needs to enter the pincode of the area where he wants to acquire the service.

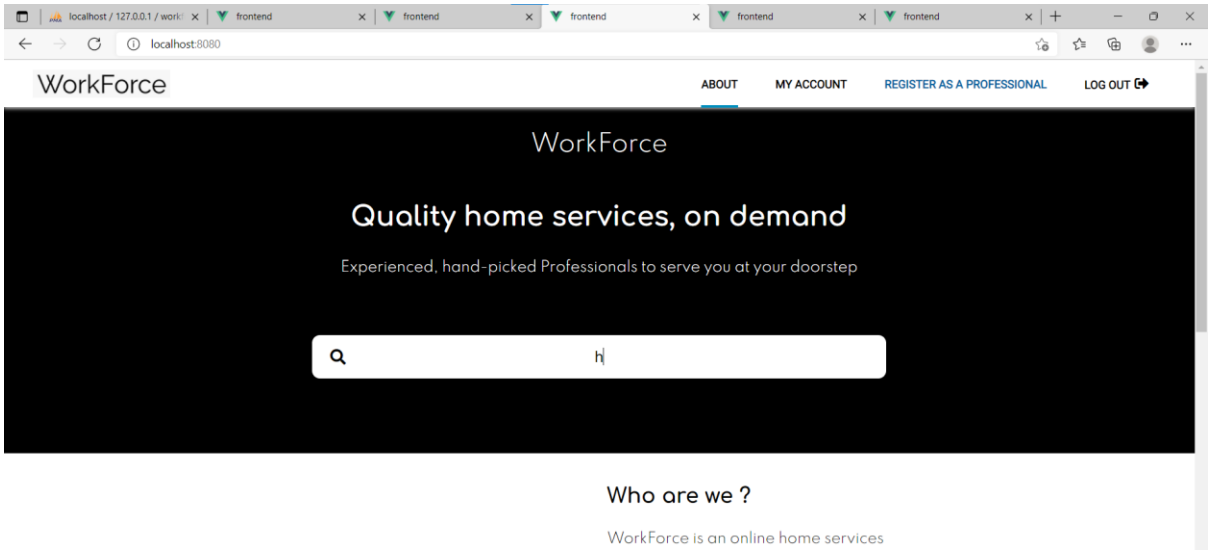



The user is now directed to a page where he needs to search for the service he wants.

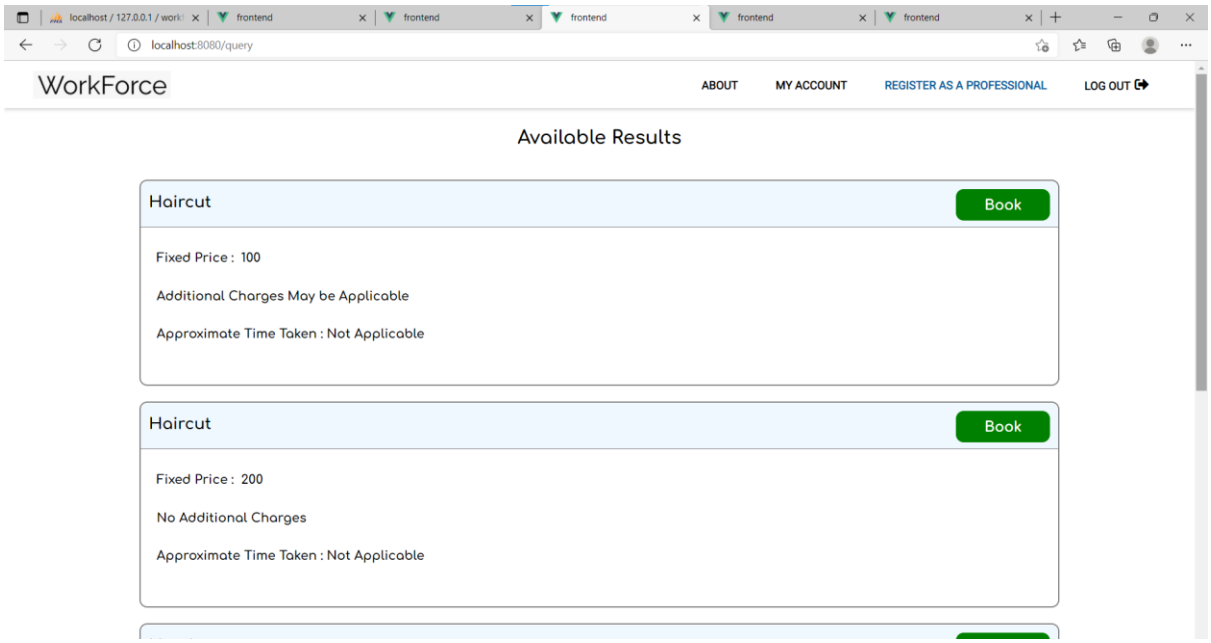


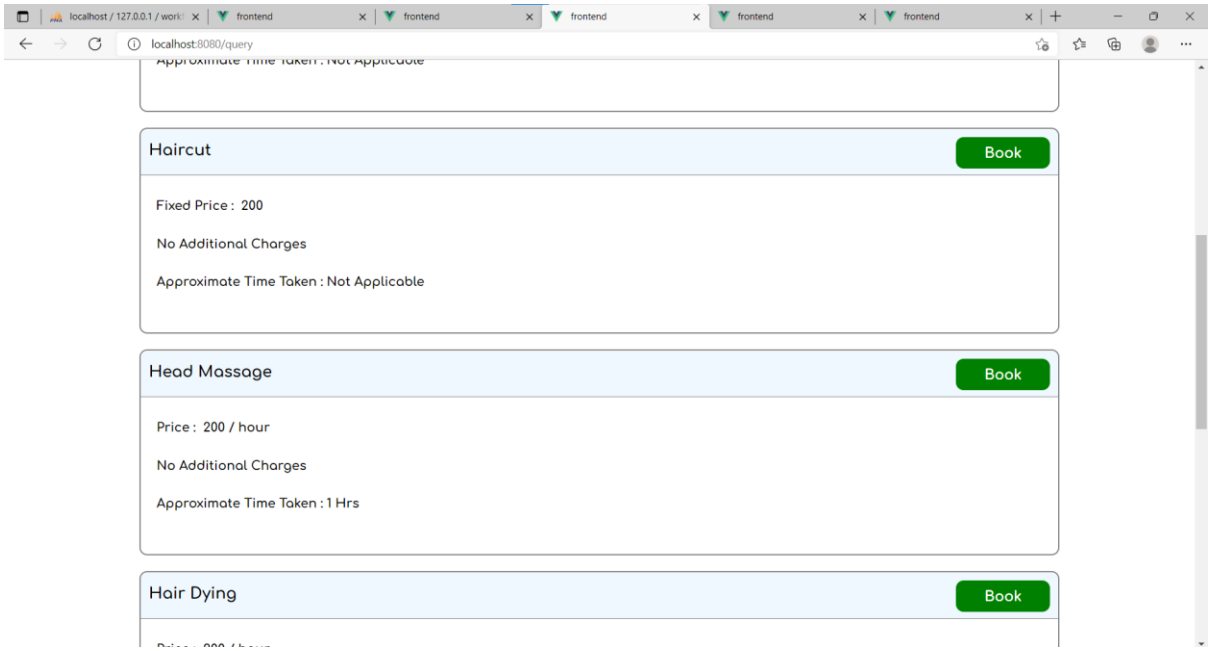

Suppose the user wants a haircut. He can just type “h” in the search box. Our website lists all the services that start with the letter “h”. When the user wants to search for a service, even entering the substring of the service name is sufficient for our website to list the service the user is looking for (this is implemented with the help of FUSE JS library).

Note: Our website also lists only the professionals in the same area as the user booking the service.

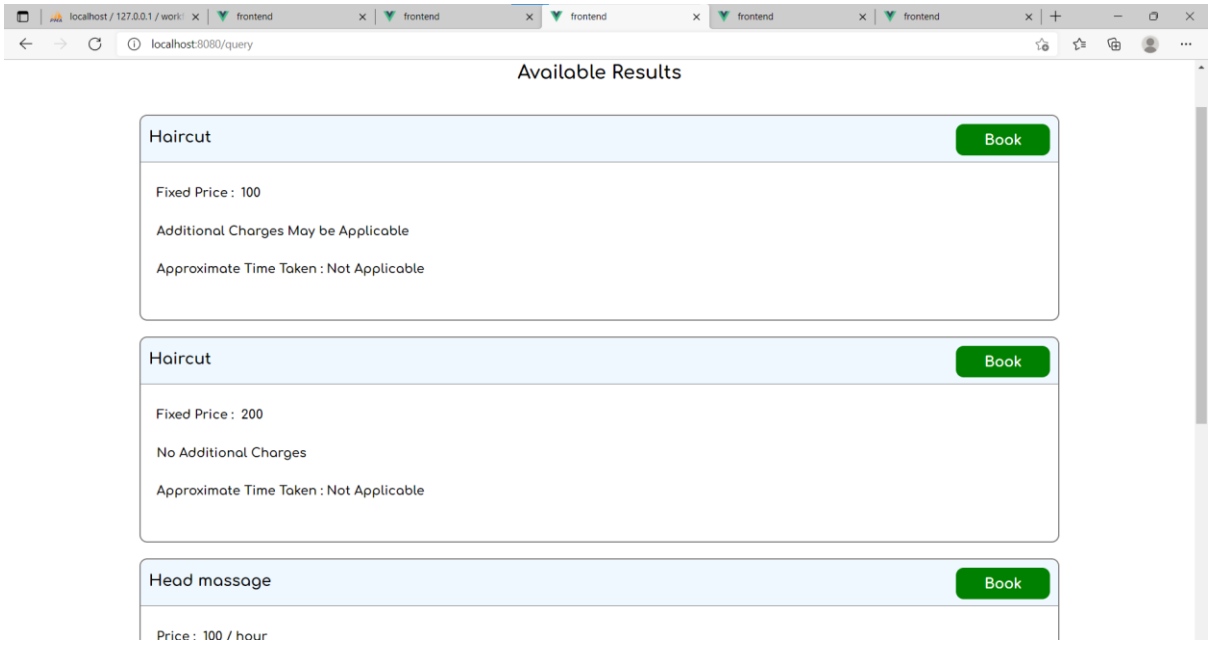


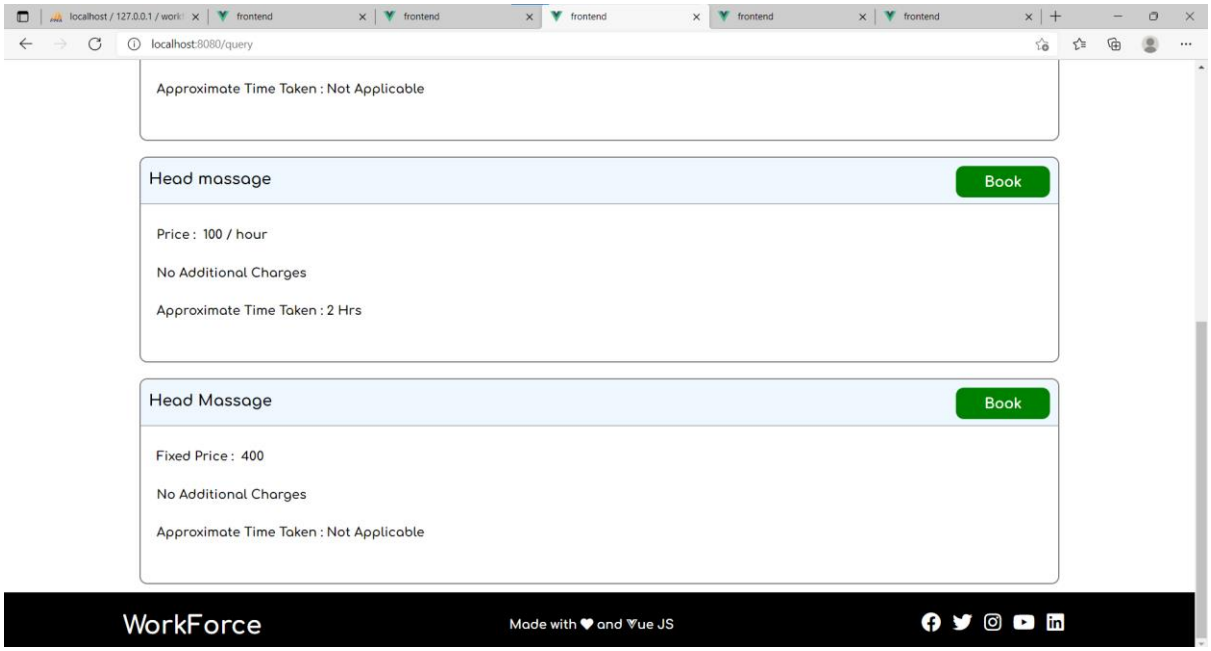

All the services (starting with the letter “h”) provided in the same area as the users’ area of residence is shown.





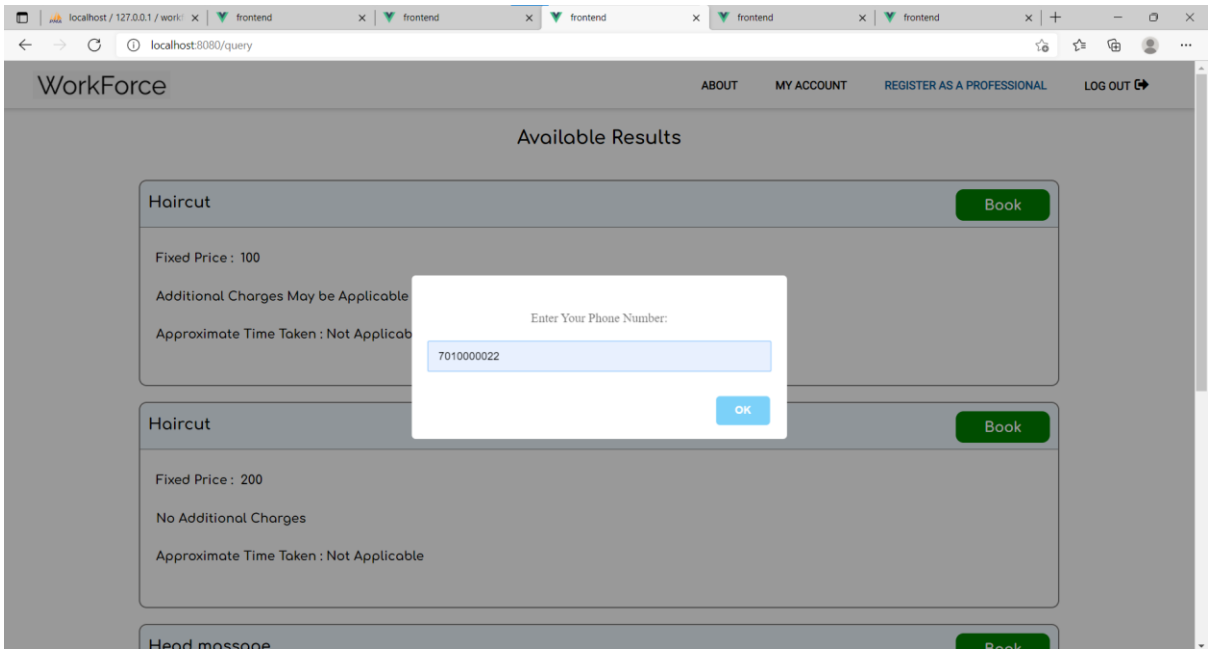

Suppose the person providing “hair dying” has now gone offline or stopped “Receiving Orders”. These are the services currently listed to the user (refer to the screenshot given below) searching for a service (in his area) that starts with the letter “h”

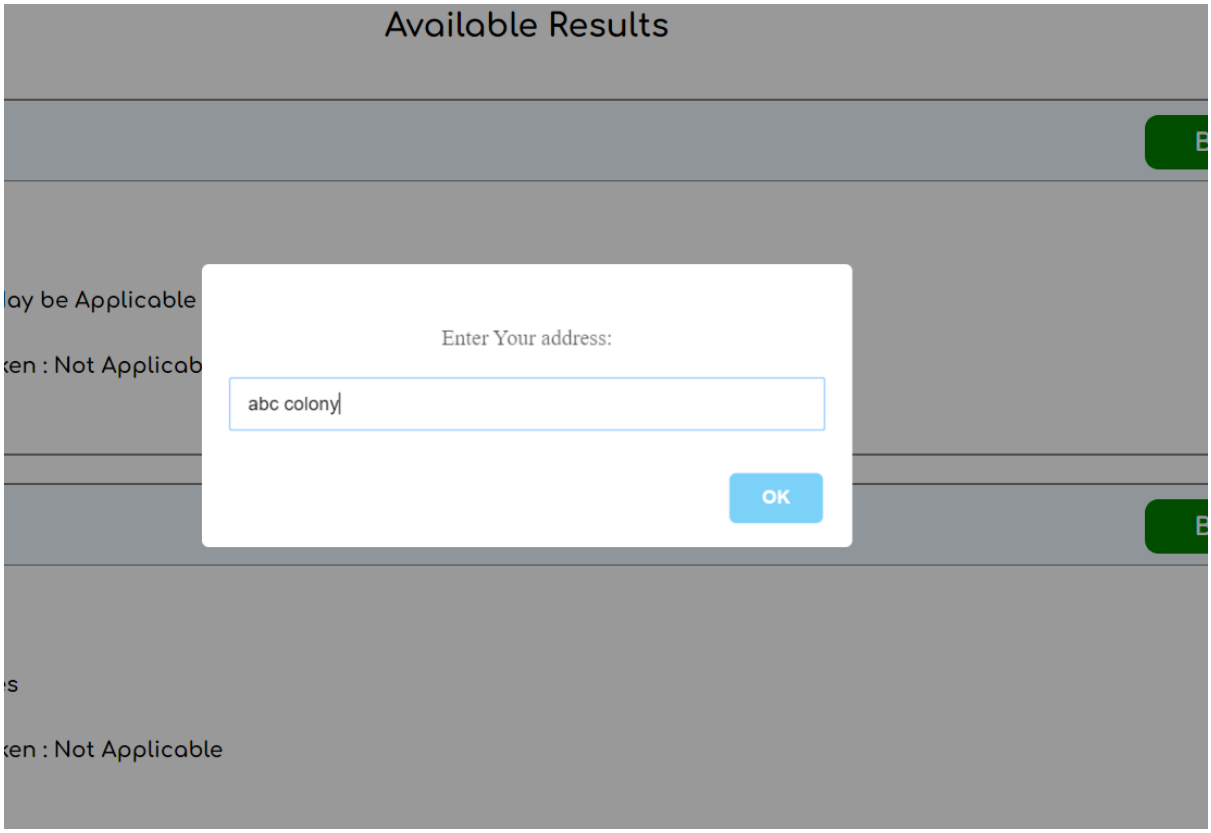





Now, the user can choose any service (in the list) offered by any professional. It may be based on the price, additional charges applied, etc. To book a service the user needs to click the “Book” button displayed beside the service list shown.

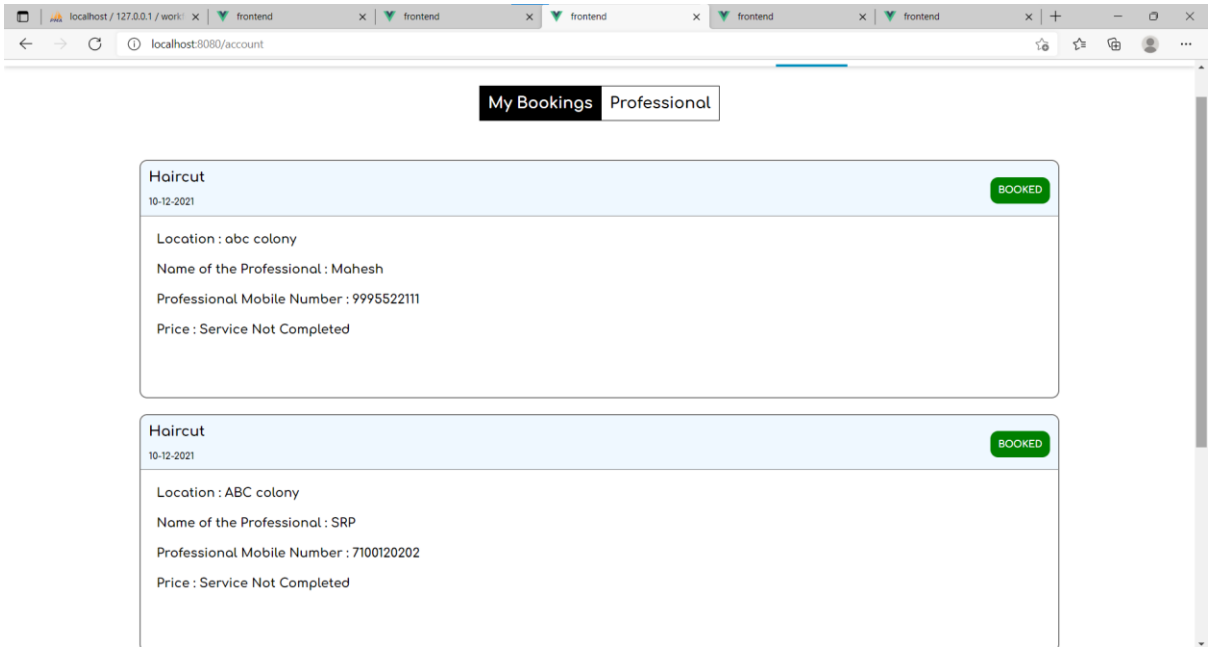
When the button “Book” is clicked, the user is asked for his mobile number and then address.

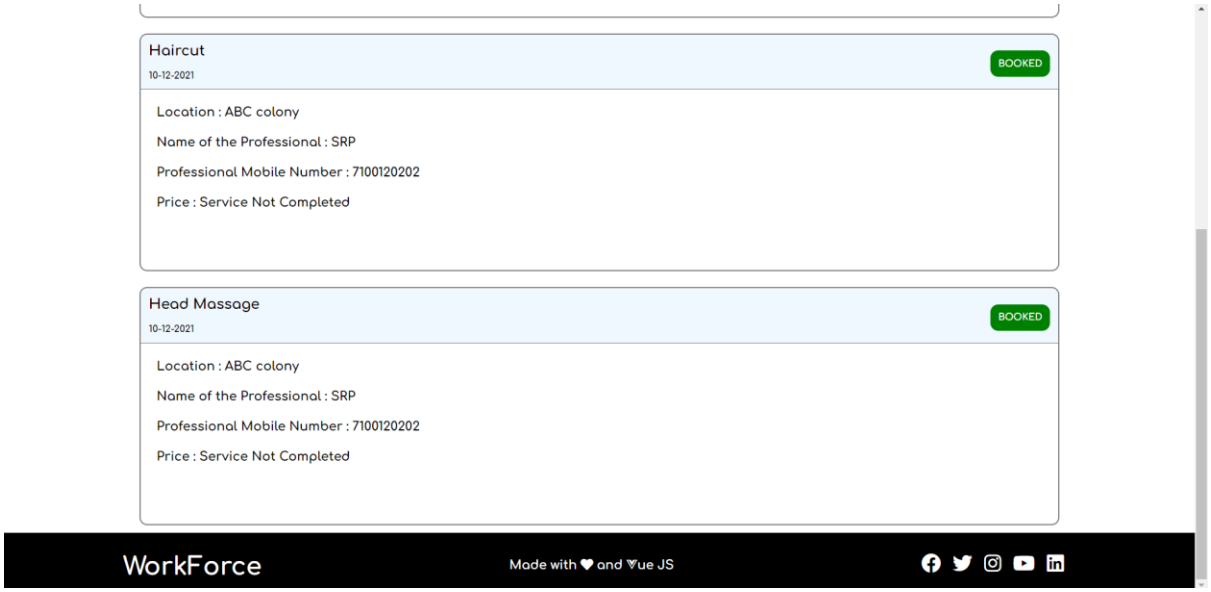





Once, the details are filled the service is booked. Let us book other services to demonstrate all functionalities of our website.

Now, when the user visits his “MY BOOKINGS” page, all his bookings are displayed. The service details are also present in it (service provider name, service provider number, etc)

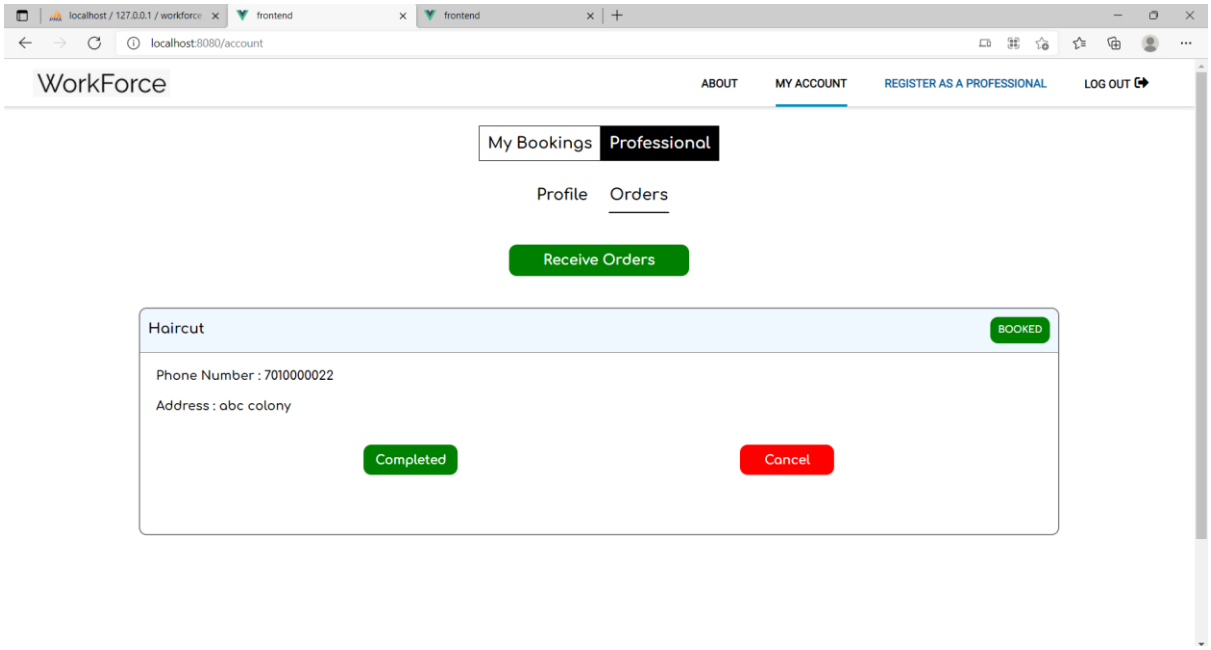




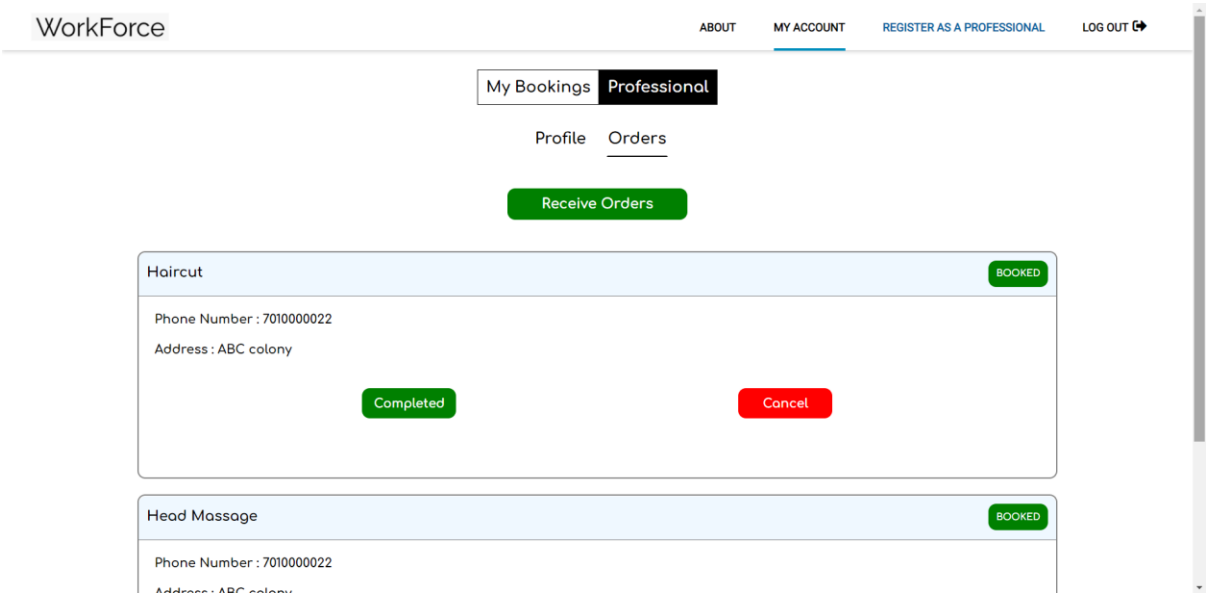

As, we can see from the screenshots above, the user has booked three services (two from SRP (service provider), one from Mahesh (service provider)

Now, let us check the My Orders page of the professionals.

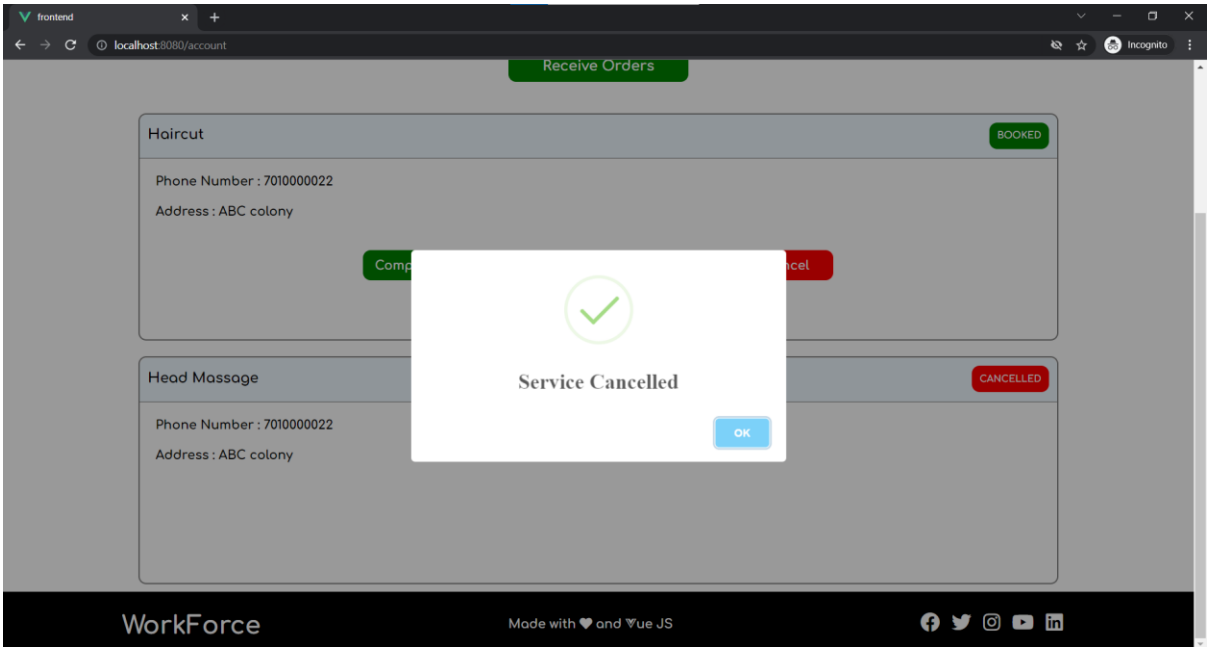
Mahesh (service provider):



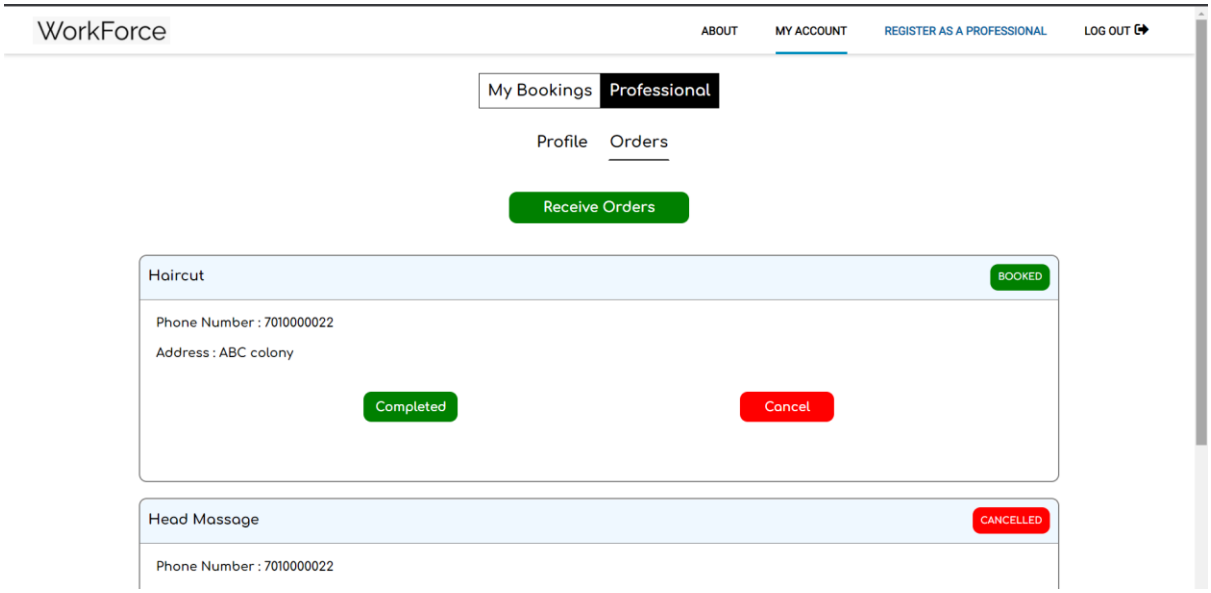
SRP (service provider)



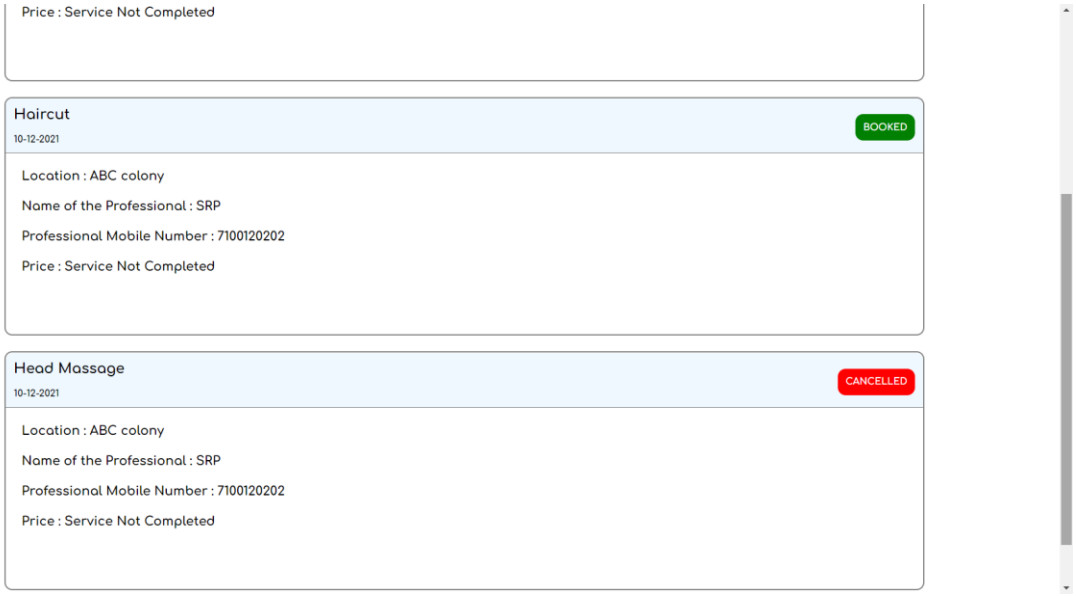

Suppose the service provider (in the case presented in the ss) wants to cancel head massage appointment. The professional needs to click the “Cancel” button present in the Orders page. When the service is cancelled by the service provider, an alert message appears on professionals’ screen claiming “Service Cancelled”



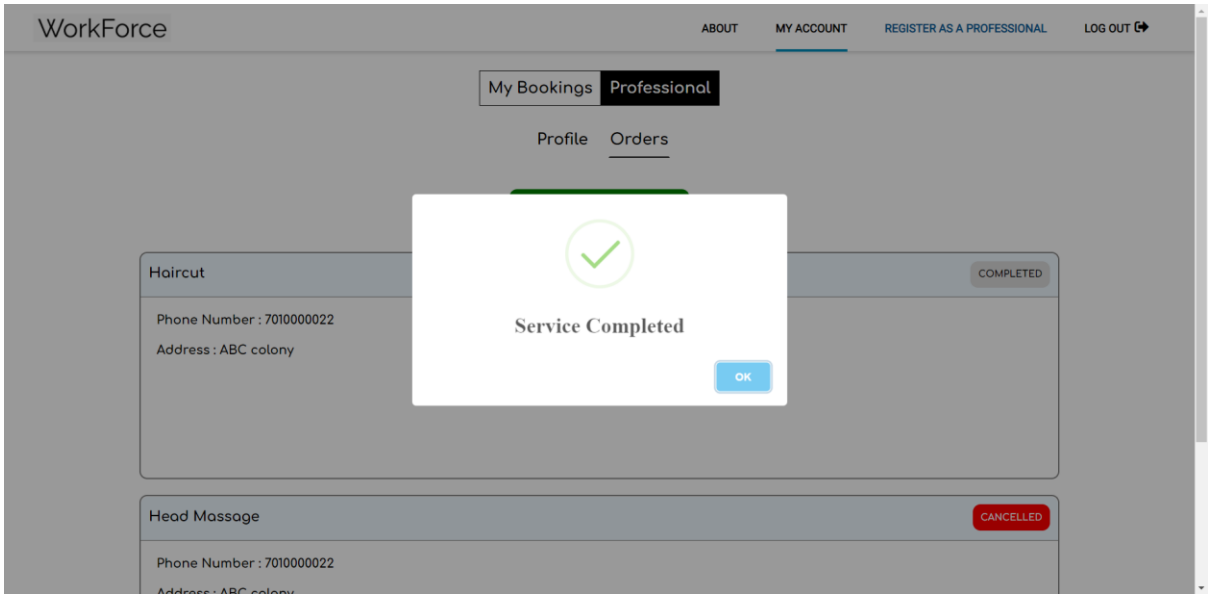

Now, when we see the orders in the professional page, we can observe that one order still remains “BOOKED” whereas the other order shows the status “CANCELLED”



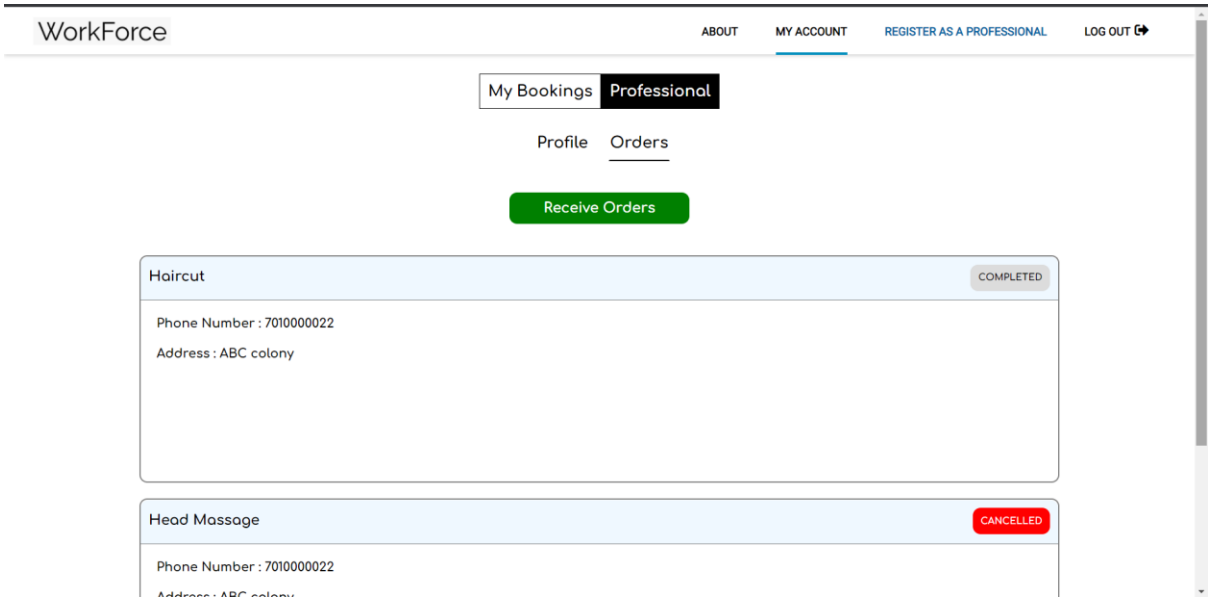

We can clearly see that the service cancelled by the professional now appears as cancelled in the users' My Bookings page too.



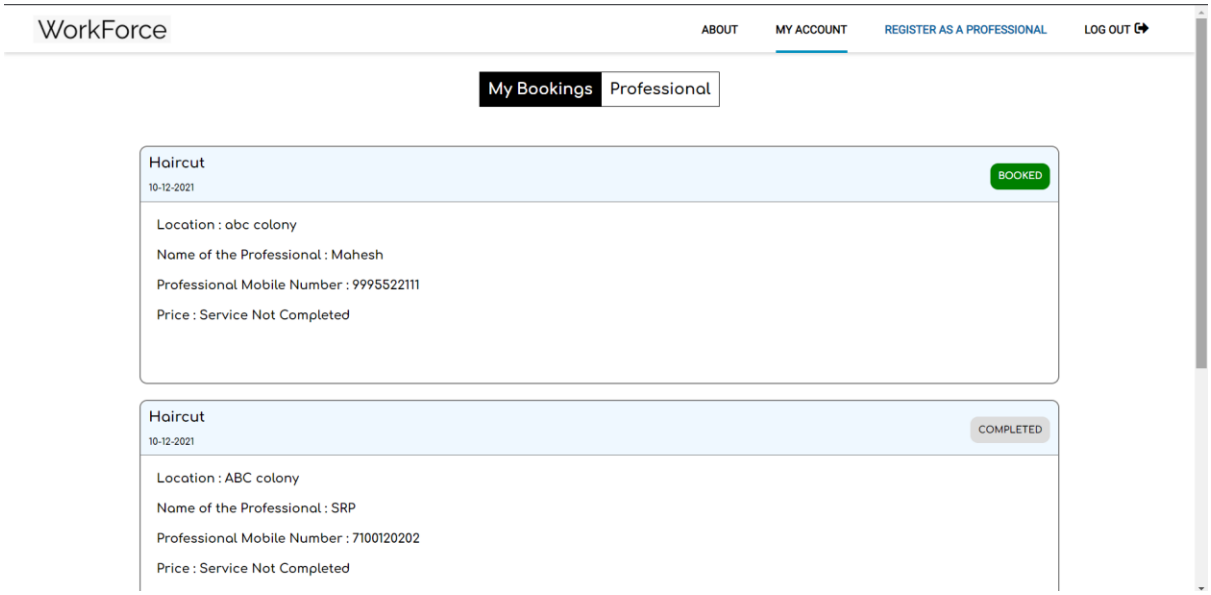

Now, suppose the professional has completed the job. Then, to denote that, the service provider needs to click on the button "COMPLETED" shown in the orders' page of a professional.



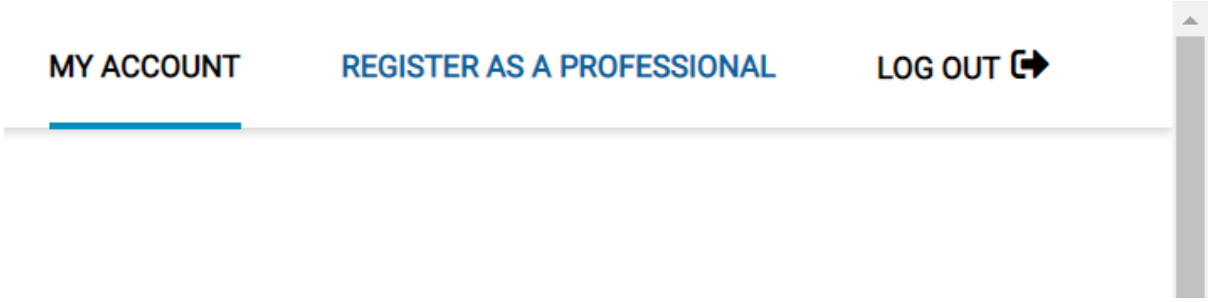

Orders page of the professional who has completed one job and cancelled another.




Users' "My Bookings" page after the professional has completed the job. Since, the professional has completed the job, we see updated results in "My Bookings" page of the user.




The user can logout from his account by clicking the “LOGOUT” button present at the top right of our website.




## CONCLUSION & FUTURE WORK:

We have created a website that acts as a platform through which skilled and experienced professionals can connect with users looking for specific services. WorkForce helps small scale business people to connect to a wider audience. People running small scale business or doing independent works have been affected the worst because of the current pandemic. They have been deprived of any new customers and also their regular ones too. These independent service providers/ people running small scale business were at the brink of becoming jobless. Our website WorkForce creates a lot of new job opportunities for the professionals registered by showing the services offered by these professional when our users search for a service they need.

For a user to register as a professional, he just needs to fill the “Register as a Professional” form. Our admin has the power to accept/deny any requests from the users who have applied to become a service provider. The professional has been given the option to

customise his page i.e, he can list out all his services and price range. The service provider receives his order only when he is online and can cancel any bookings he has if he feels that his workload is high. A distinct feature provided by WorkForce is that our customers can choose the service provider i.e, a customer can choose any registered professional providing the service listed in our website (under customer's area).

For future works, we have thought of some functionalities that can be added to our website. One such functionality is asking for users' feedback on the work done by the professional after every service. This may help our website to ensure that the service providers listed by us continue to provide quality services to our trusted customers.

## REFERENCES:

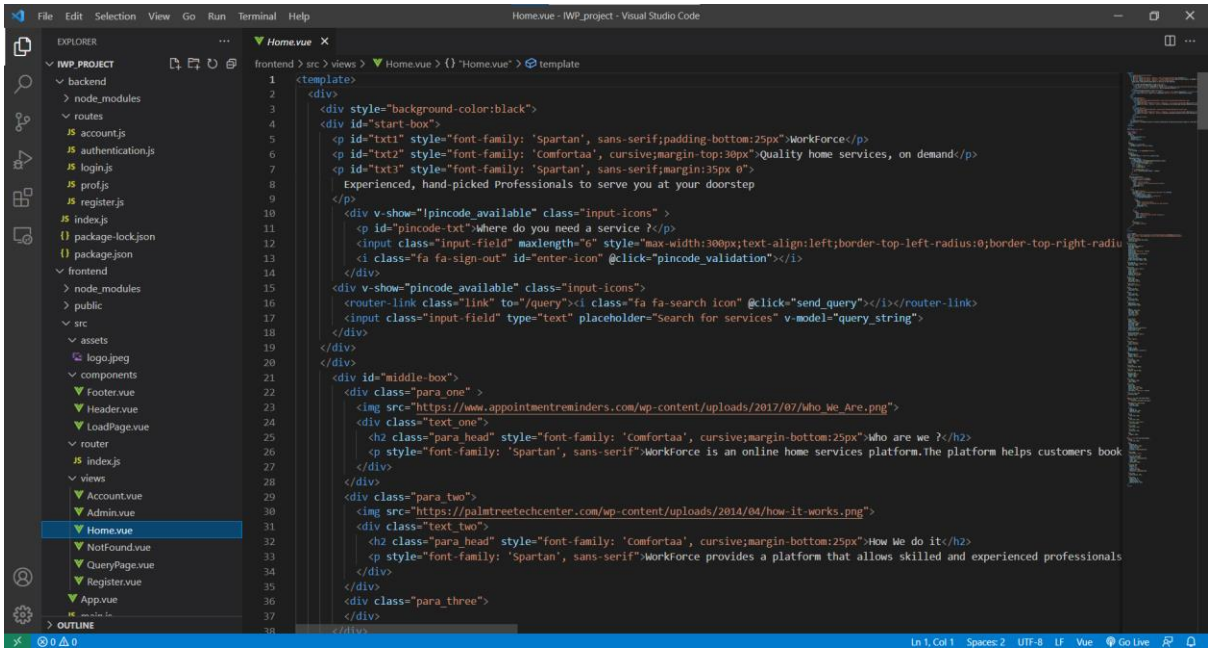

- https://vuejs.org/
- https://vuejs.org/v2/guide/installation.html
- https://www.urbancompany.com/
- https://fusejs.io/
- https://sweetalert.js.org/guides/
- https://sweetalert2.github.io/
- https://www.tutorialspoint.com/vuejs/vuejs_overview.htm
- https://stefankrause.net/js-frameworks-benchmark4/webdriver-ts/table.html
- https://angular.io/


## LINK TO THE PROJECT REPOSITORY:

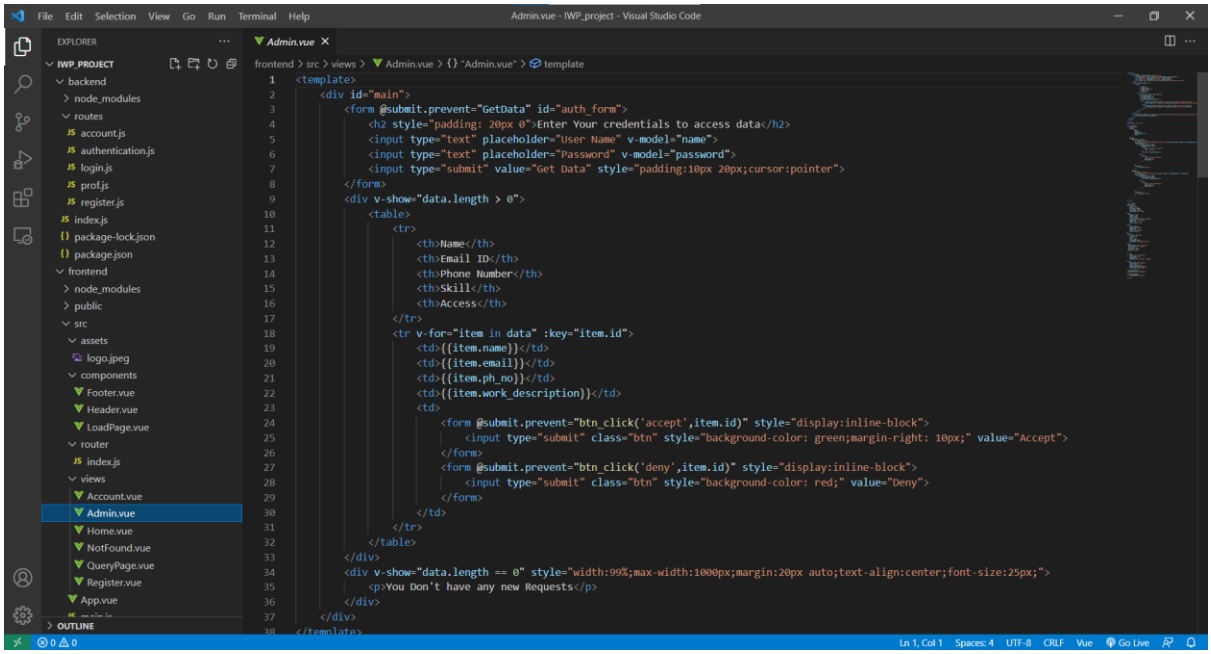
https://github.com/praveensr1202/IWP-PROJECT

# CODE SNIPPETS:

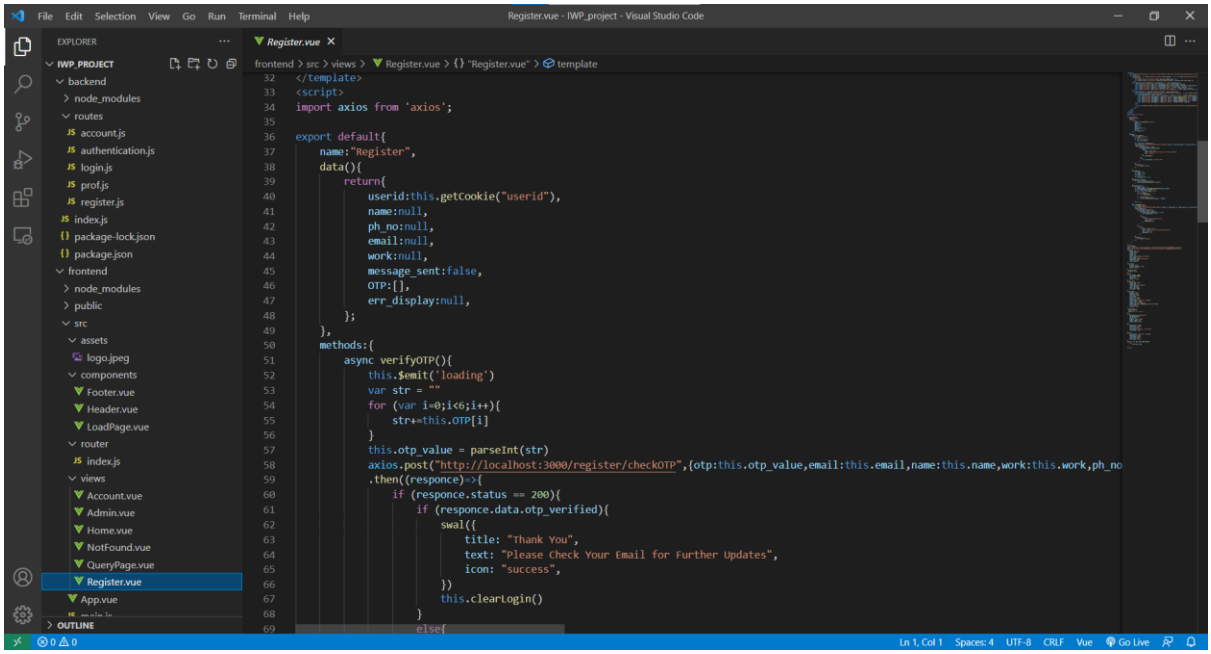
Home.vue



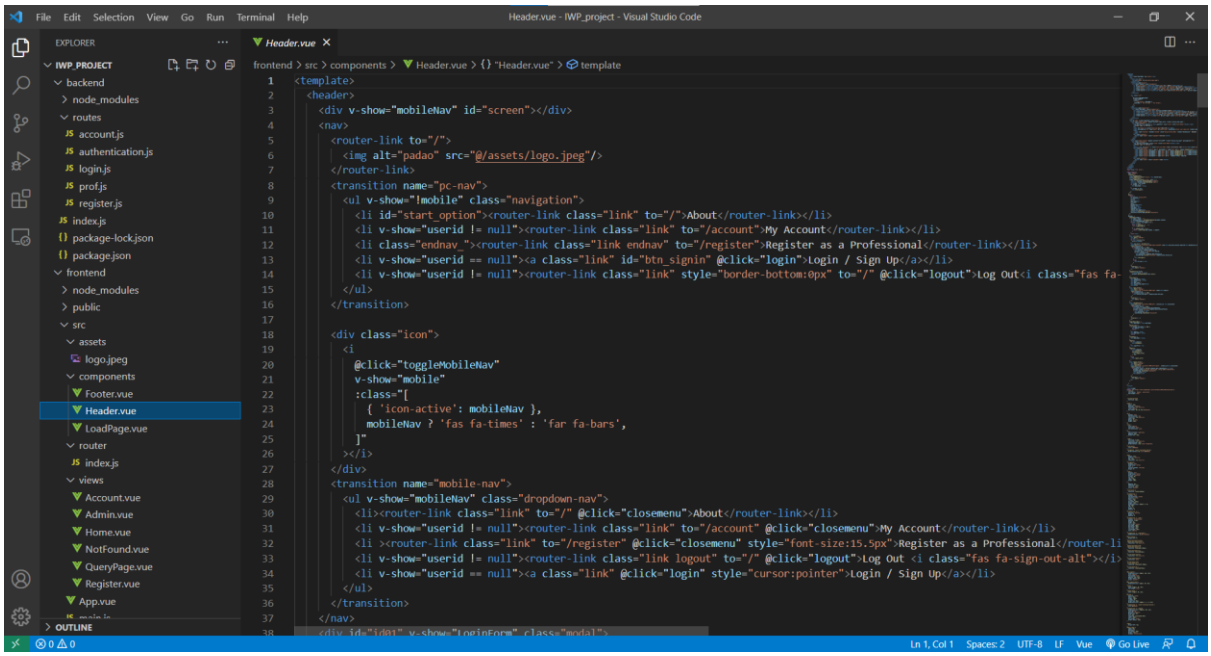

Admin.vue



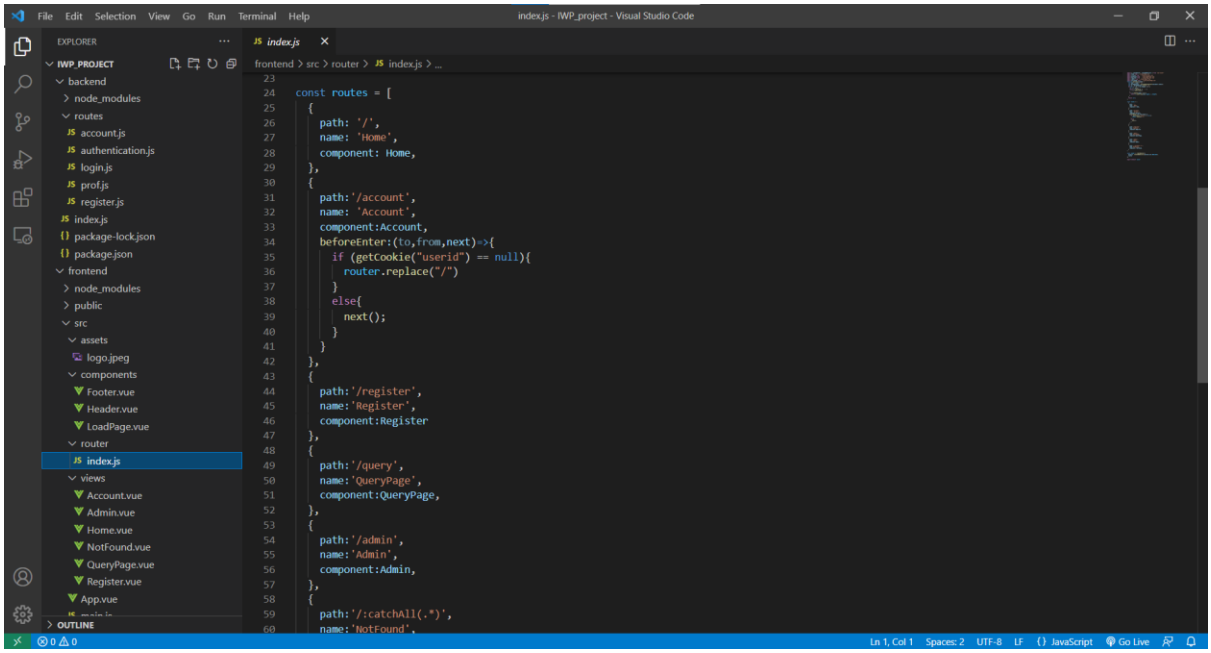

Script in Register.vue

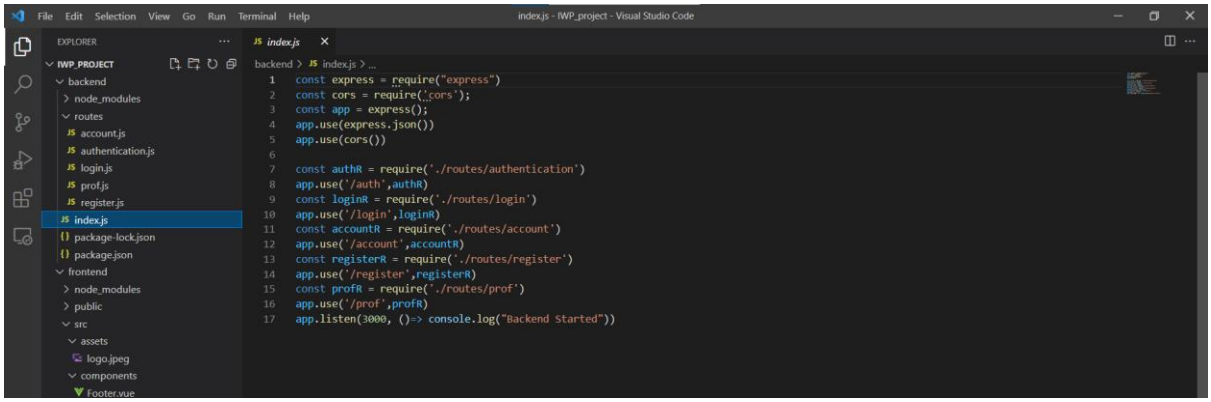



Header page code ss:

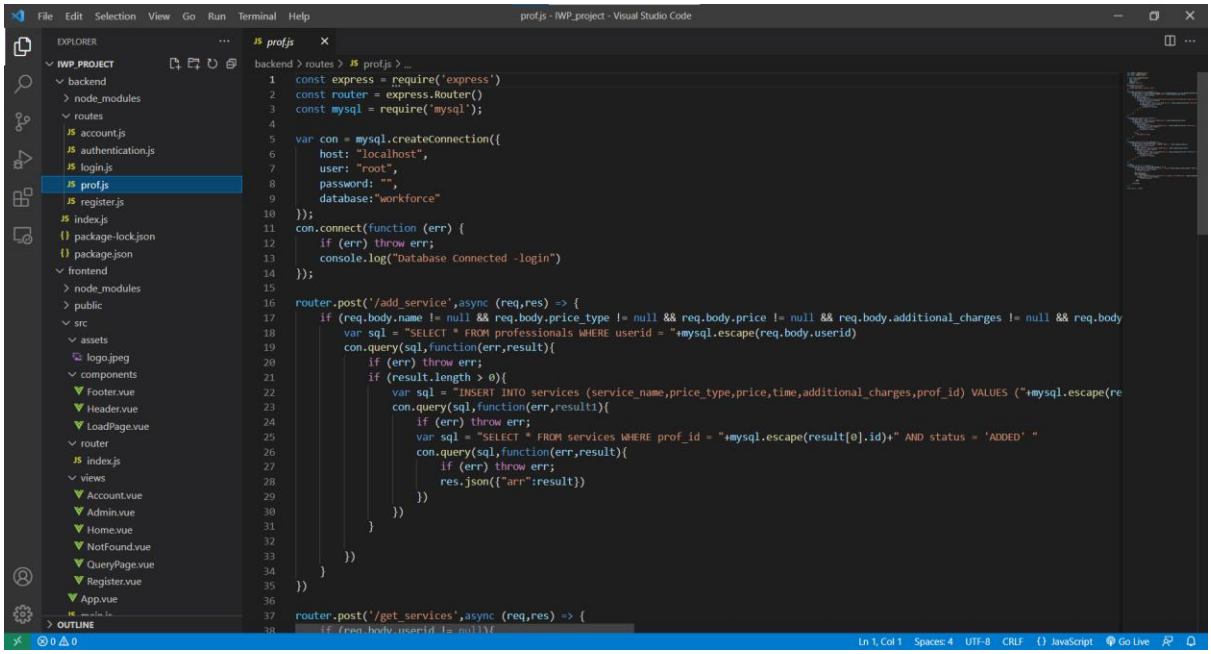



Routes to other pages:

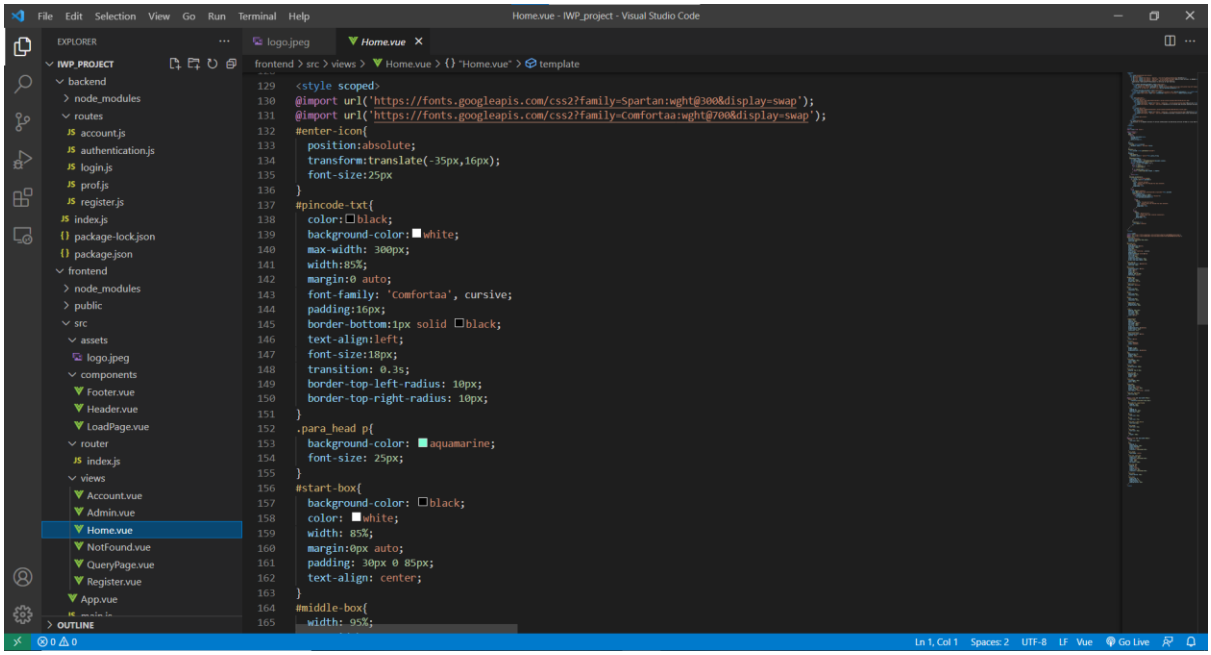

File Edit Selection View Go Run Terminal Help
index.js - IWP_project - Visual Studio Code
EXPLORER
IWP_PROJECT
backend
node_modules
routes
account.js
authentication.js
login.js
prof.js
register.js
index.js
package-lock.json
package.json
frontend
node_modules
public
src
assets
logo.jpeg
components
Footer.vue
Header.vue
LoadPage.vue
router
index.js
views
Account.vue
Admin.vue
Home.vue
NotFound.vue
QueryPage.vue
Register.vue
App.vue
OUTLINE
index.js
frontend > src > router > index.js > ...
const routes = [
{
path: '/',
name: 'Home',
component: Home,
},
{
path:'/account',
name: 'Account',
component:Account,
beforeEnter:(to,from,next)=>{
if (getCookie("userid") == null){
router.replace("/")
}
else{
next();
}
}
},
{
path:'/register',
name:'Register',
component:Register
},
{
path:'/query',
name:'QueryPage',
component:QueryPage,
},
{
path:'/admin',
name:'Admin',
component:Admin,
},
{
path:'/:catchAll(.*)',
name:'NotFound',
Ln 1, Col 1
Spaces: 2
UTF-8
LF
JavaScript
Go Live


File Edit Selection View Go Run Terminal Help
index.js - IWP_project - Visual Studio Code
EXPLORER
IWP_PROJECT
backend
node_modules
routes
account.js
authentication.js
login.js
prof.js
register.js
index.js
package-lock.json
package.json
frontend
node_modules
public
src
assets
logo.jpeg
components
Footer.vue
index.js
backend > index.js > ...
const express = require("express")
const cors = require('cors');
const app = express();
app.use(express.json())
app.use(cors())
const authR = require('./routes/authentication')
app.use('/auth',authR)
const loginR = require('./routes/login')
app.use('/login',loginR)
const accountR = require('./routes/account')
app.use('/account',accountR)
const registerR = require('./routes/register')
app.use('/register',registerR)
const profR = require('./routes/prof')
app.use('/prof',profR)
app.listen(3000, ()=> console.log("Backend Started"))

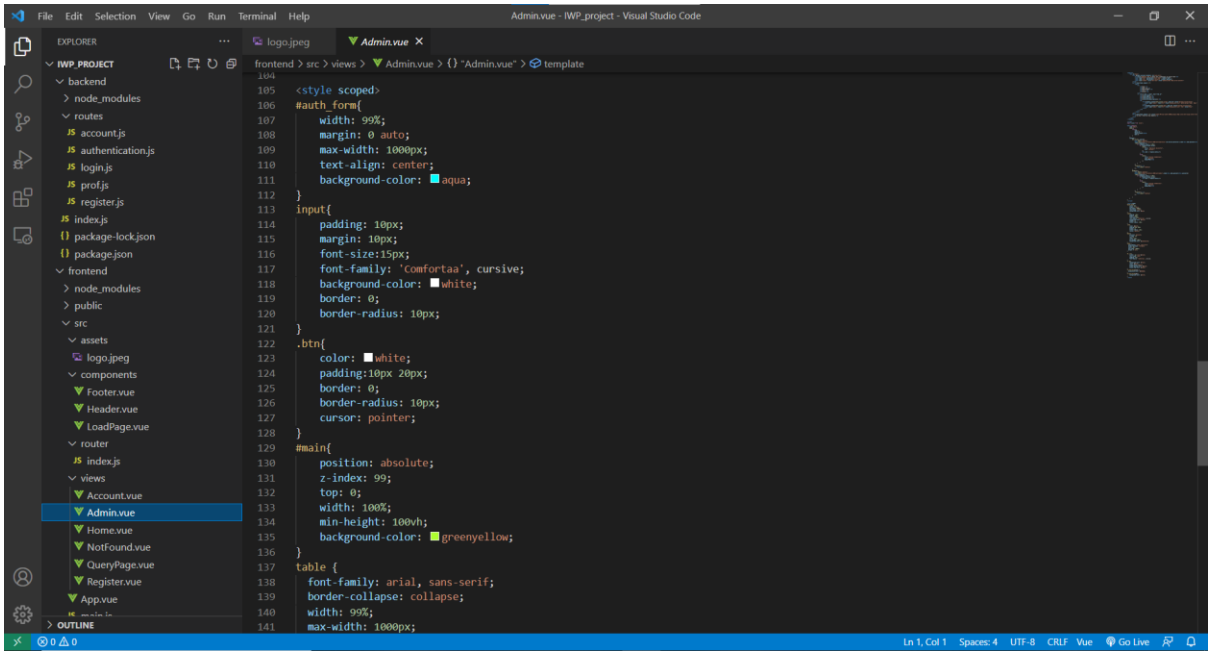
prof.js




Home page styling:




Admin page styling:


File Edit Selection View Go Run Terminal Help
Admin.vue - IWP_project - Visual Studio Code
EXPLORER
IWP_PROJECT
backend
node_modules
routes
account.js
authentication.js
login.js
prof.js
register.js
index.js
package-lock.json
package.json
frontend
node_modules
public
src
assets
logo.jpeg
components
Footer.vue
Header.vue
LoadPage.vue
router
index.js
views
Account.vue
Admin.vue
Home.vue
NotFound.vue
QueryPage.vue
Register.vue
App.vue
OUTLINE
logo.jpeg
Admin.vue
frontend > src > views > Admin.vue > {} "Admin.vue" > template
104
105 <style scoped>
106 #auth_form{
107 width: 99%;
108 margin: 0 auto;
109 max-width: 1000px;
110 text-align: center;
111 background-color: aqua;
112 }
113 input{
114 padding: 10px;
115 margin: 10px;
116 font-size:15px;
117 font-family: 'Comfortaa', cursive;
118 background-color: white;
119 border: 0;
120 border-radius: 10px;
121 }
122 .btn{
123 color: white;
124 padding:10px 20px;
125 border: 0;
126 border-radius: 10px;
127 cursor: pointer;
128 }
129 #main{
130 position: absolute;
131 z-index: 99;
132 top: 0;
133 width: 100%;
134 min-height: 100vh;
135 background-color: greenyellow;
136 }
137 table {
138 font-family: arial, sans-serif;
139 border-collapse: collapse;
140 width: 99%;
141 max-width: 1000px;
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Vue Go Live