**Technical Documentation**

**Handyman**

**Team Members:**

|  |  |
| --- | --- |
| **Name** | **Student #** |
| Abby Shah | 300759250 |
| Neeti Parekh | 301000838 |
| Bohdan Kucher | 300967325 |
| Helia Mozaffari | 300824698 |
| Lee Soukhee (Billy) | 300873157 |
| Jaesoon Lee | 300914208 |
| Imtiaz Khan | 300922598 |

[**Introduction**](#_uooz10a6ce2g) **3**

[In this document you will find](#_nlc36ypipj7b) 3

[Intended Audience](#_9gz7p2ypgsxc) 3

[Required knowledge and skills](#_216myi6b1ll9) 3

[How to use this document](#_jc6gtlfrsgj0) 3

[**Chapter 1**](#_9lrsbxr4gjn9) **4**

[How to start the Project](#_lcrcwcwa6qnc) 4

[Released Features](#_gl4gmc76ivd4) 8

[**Chapter 2**](#_szyrxlig4smc) **8**

[Handyman Breakdown Structure](#_77uhzmjanucu) 8

[ERD Diagram](#_wmerie3xtjmq) 9

[API](#_h263o627hlb5) 10

[**Chapter 3**](#_dxljcme4ebp) **14**

[Testing APIs](#_i8qawvum6f31) 14

[Server Availability check](#_w8s4q7vzakhc) 15

# 

# Introduction

Handyman is an application for selling handyman services, this application developed using Vue.JS.

## In this document you will find

* How to start the project
* Release Features
* Handyman Breakdown Structure
* ERD
* API requests and responses
* Testing APIs

## Intended Audience

* This Document intended for developers who will be integrating maintaining and updating handyman application

## Required knowledge and skills

* Knowledge of Vue.Js platform
* Knowledge of API development and consumption
* Software development skills

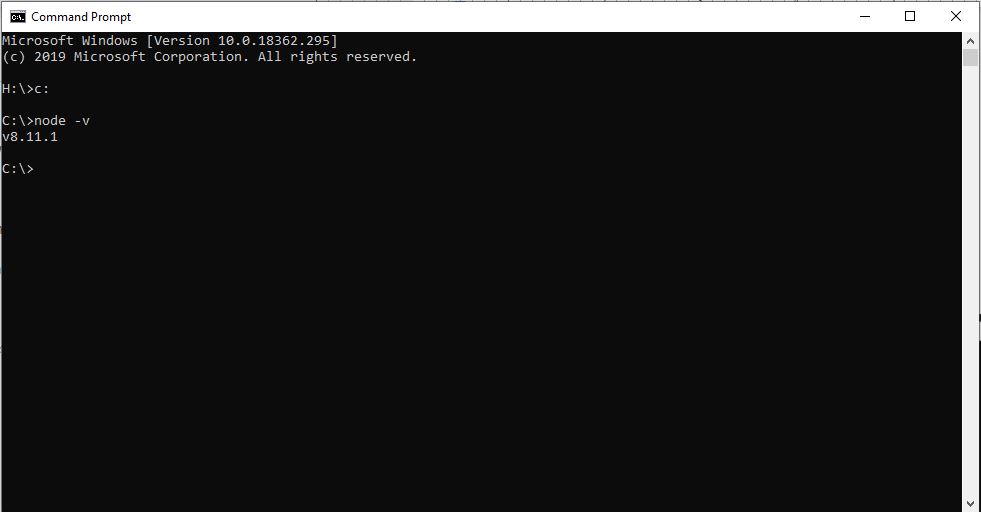
## How to use this document

* If this is your first time developing handyman software please ask the scrum master to send you an invitation to have access to the project through Github and follow the steps indicated in chapter one (How to start the project).
* If you want to familiarize yourself with the project please check Handyman Top Level diagram and Released features.
* If you are developing the backend of this application database schema is available under in second chapter (ERD)
* If you are developing the middle layer or consuming the API use the second chapter (API requests and responses)
* If you are Testing the consumption of the APIs please refer to chapter 3 (Testing APIs)

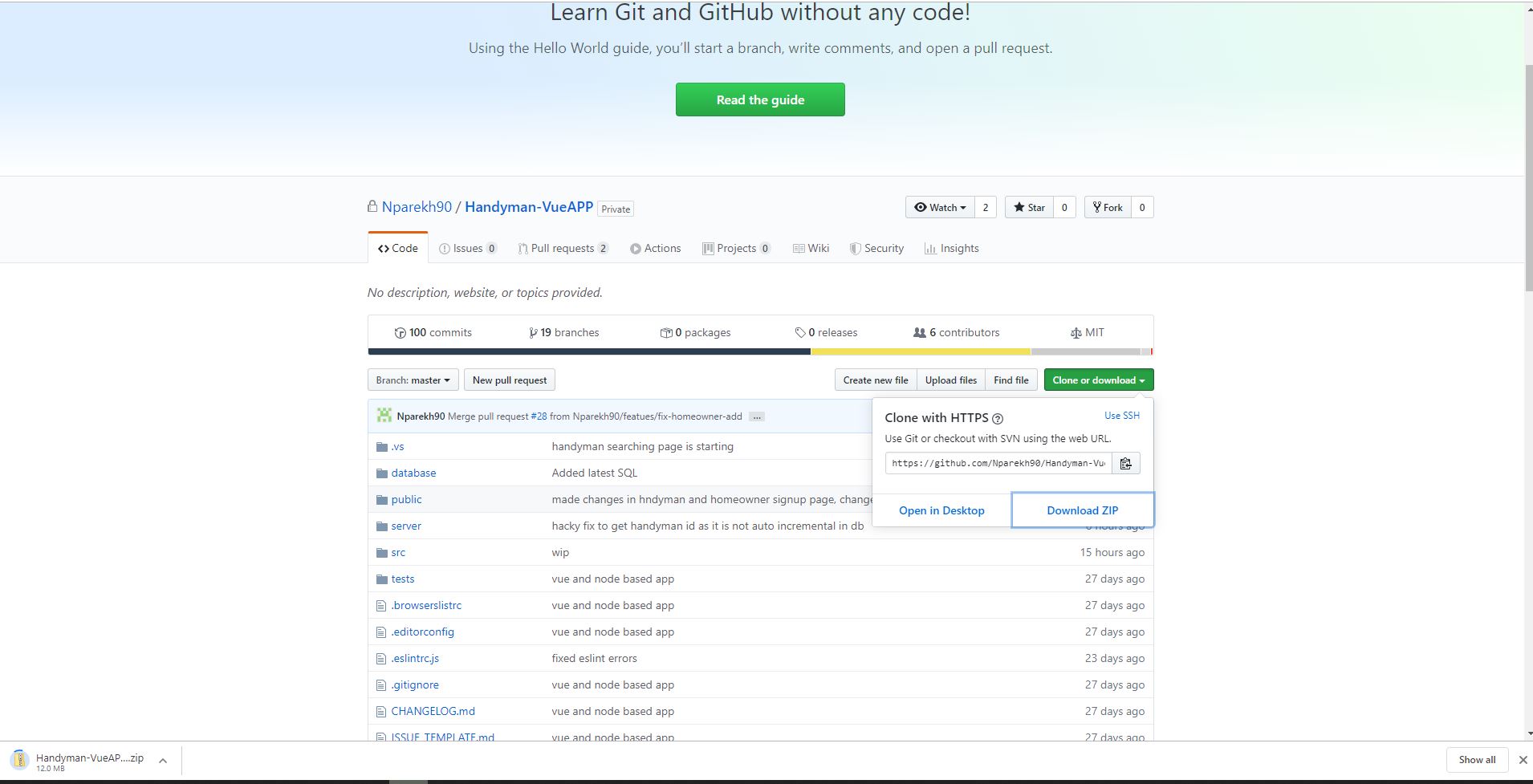
# Chapter 1

## How to start the Project

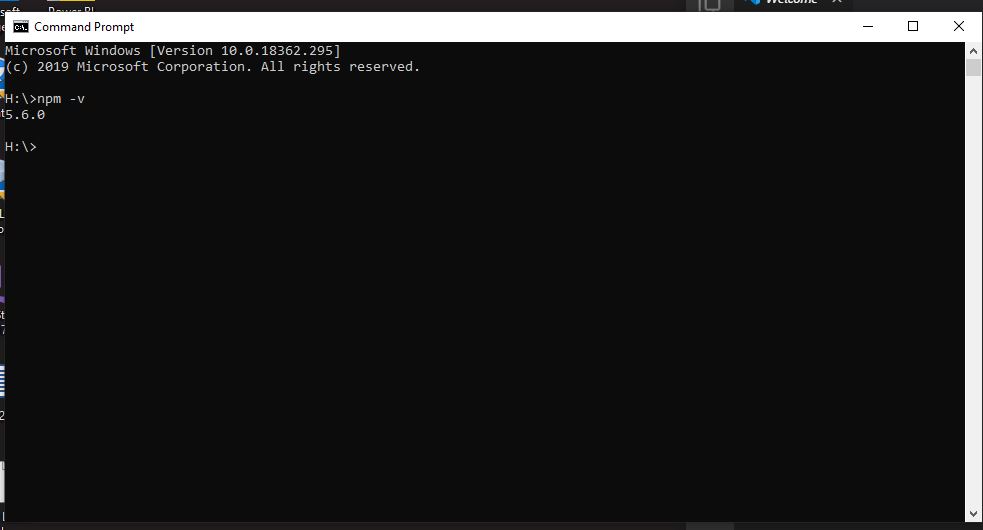
1. Install [Visual Studio Code](https://code.visualstudio.com/)
2. Install [Node.js](https://nodejs.org/en/download/) (This installation includes npm and chocolatey)
3. Validate installations



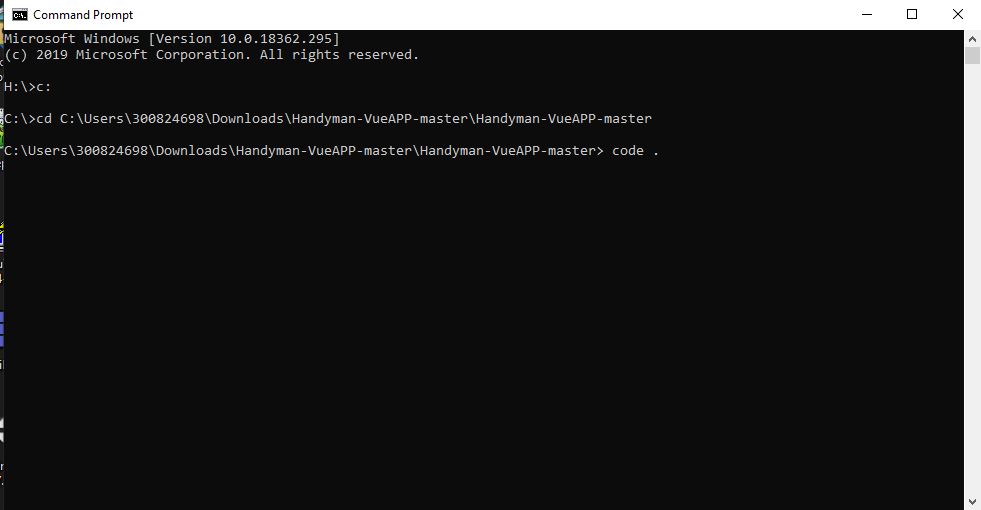
1. Clone [this Repository](https://help.github.com/en/desktop/contributing-to-projects/cloning-a-repository-from-github-desktop) onto your PC



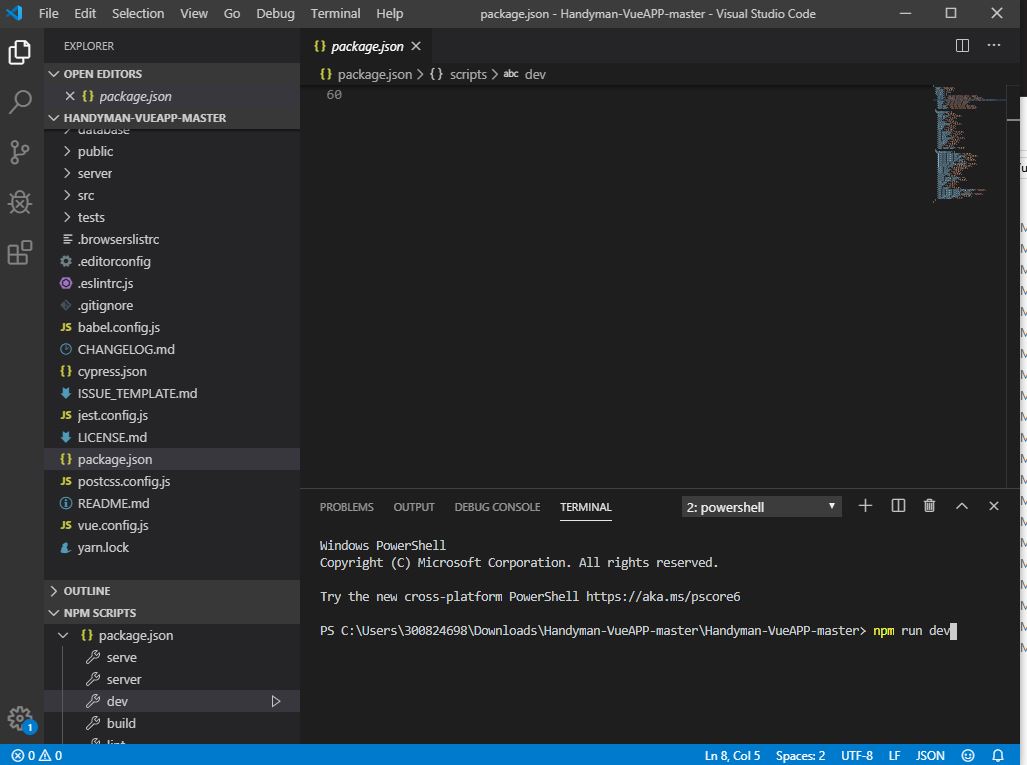
1. Inside the local repository root folder, run ‘npm -v’ to check the version and if it was not install write ‘npm install’. This will install the required dependencies of the project.



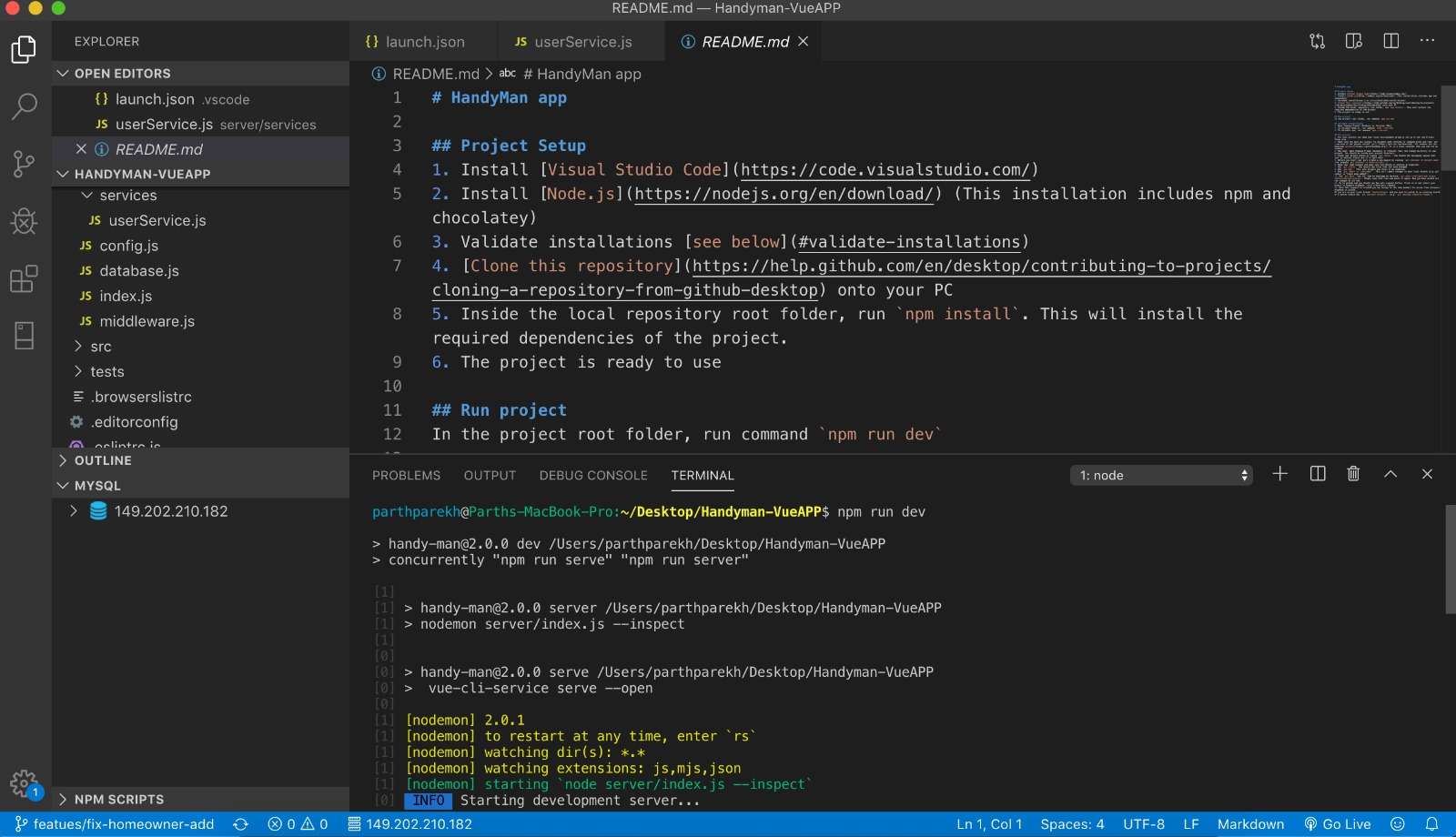
1. In terminal write cd “path to the project folder”.
2. Write “code .” and visual code will open with the project



1. Open a new terminal in visual code
2. In the terminal run command `npm run dev`



1. You should see this in visual code and the homepage of the website should open.

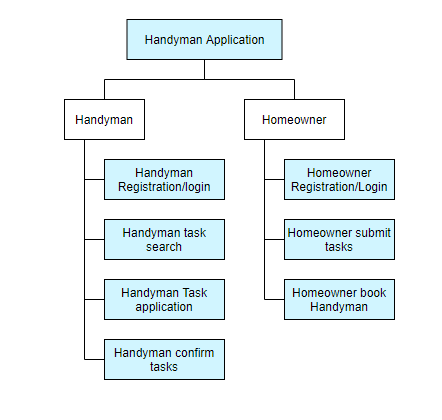


## Released Features

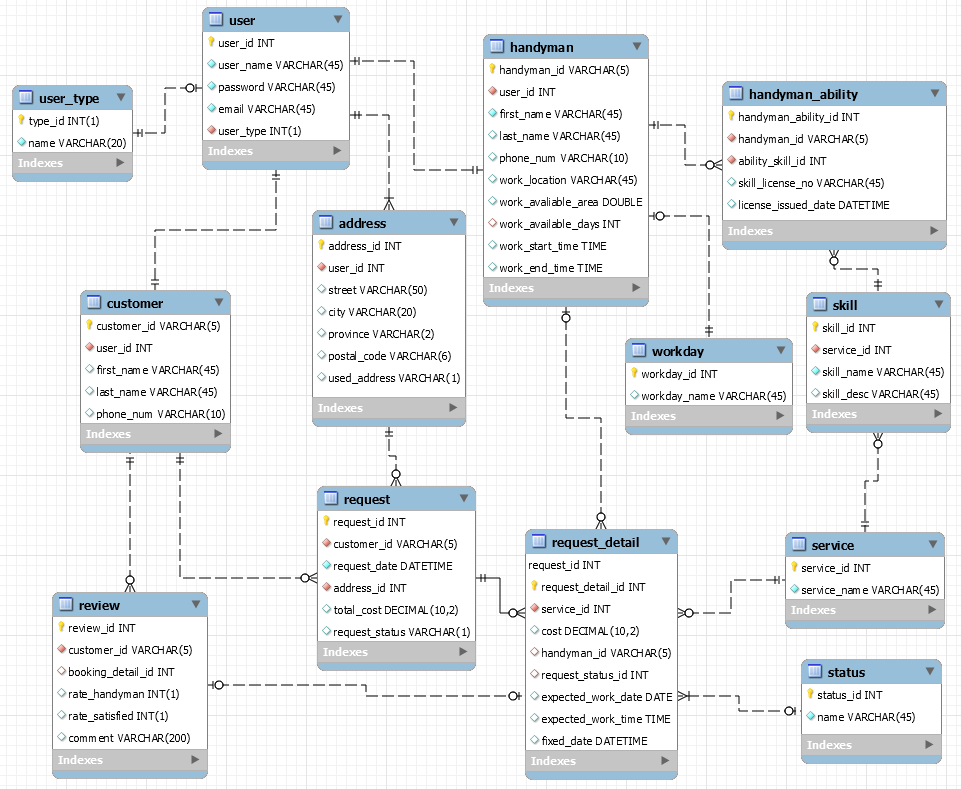
|  |
| --- |
| **November 2019** |
| Handyman Registration and login |
| Handyman search |
| Book a Handyman |
| Token for session storage |

# Chapter 2

## Handyman Breakdown Structure



## ERD

****

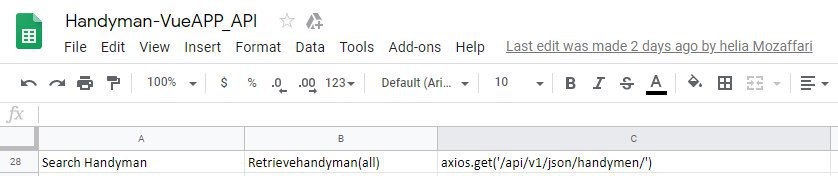
## API

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Web Pages** | **Action** | **Call from frontend** | **Params** | **Sources** | **Remarks** |
| Login | Login | axios.get('/api/v1/json/users/finduser') | userName, password | Server/routes/users |  |
| Register for HomeOwner | Register or Save | axios.post('/api/v1/json/users/add') | userName, password, email, userType | Server/routes/users |  |
|  |  | axios.post('/api/v1/json/adresses/add') | userID, street, city, province, postalCode, usedAddress | Server/routes/adresses |  |
|  |  | axios.post('/api/v1/json/customers/add') | userID, street, city, province, postalCode, usedAddress | Server/routes/customers |  |
| Register for Handyman | Register or Save | axios.post('/api/v1/json/users/add') | userName, password, email, userType | Server/routes/users |  |
|  |  | axios.post('/api/v1/json/adresses/add') | userID, street, city, province, postalCode, usedAddress | Server/routes/adresses |  |
|  |  | axios.post('/api/v1/json/handymen/add') | userID, firstName, lastName, phoneNum, workLocation,  workArea, workDays, work, workStartTime, workEndTime | Server/routes/handymen |  |
|  |  | axios.post('/api/v1/json/handymanAbilities/add') | handymanID, skillID, skill\_licenseNo, issuedDate | Server/routes/handymanAbilities |  |
|  | Retrieve skill list | axios.get('/api/v1/json/skills/') | skill\_id, service\_id, skill\_name,. skill\_desc | Server/routes/skills | refer Skill\_SampleData Sheet |
| Update profile for HomeOwner | RetrieveProfile | axios.get('/api/v1/json/users/'+userID) | userID, firstName, LastName | Server/routes/users |  |
|  |  | axios.get('/api/v1/json/addresses/'+userID) | Address, street, city, province, postal\_code | Server/routes/adresses |  |
|  |  | axios.get('/api/v1/json/customers/'+customerID) |  | Server/routes/customers |  |
|  | Update profile | axios.put('/api/v1/json/users/edit/'+userID) | password | Server/routes/users |  |
|  |  | axios.put('/api/v1/json/addresses/edit/'+userID) | street, city, province, postalCode, usedAddress | Server/routes/adresses |  |
|  |  | axios.put('/api/v1/json/customers/edit/'+customerID) | firstName, lastName, phoneNum | Server/routes/handymen |  |
|  | Delete address | axios.delete('/api/v1/json/addresses/delete/'+userID) |  | Server/routes/handymanAbilities |  |
| Update profile for Handyman | RetrieveProfile | axios.get('/api/v1/json/users/'+userID) |  | Server/routes/users |  |
|  |  | axios.get('/api/v1/json/addresses/'+userID) |  | Server/routes/adresses |  |
|  |  | axios.put('/api/v1/json/handymen/'+handmanID) |  | Server/routes/handymen |  |
|  |  | axios.put('/api/v1/json/handymanAbilities/'+handmanID) |  | Server/routes/handymanAbilities |  |
|  | Update profile | axios.put('/api/v1/json/users/edit/'+userID) | password | Server/routes/users |  |
|  |  | axios.put('/api/v1/json/addresses/edit/'+userID) | street, city, province, postalCode, usedAddress | Server/routes/adresses |  |
|  |  | axios.put('/api/v1/json/handymen/edit/'+handmanID) | lastName, phoneNum, workLocation, workArea,  workDays, workStartTime, workEndTime | Server/routes/handymen |  |
|  |  | axios.put('/api/v1/json/handymanAbilities/edit/'+handmanID) | handymanID, skillID, skill\_licenseNo, issuedDate | Server/routes/handymanAbilities |  |
|  | Delete address | axios.delete('/api/v1/json/addresses/delete/'+userID) |  | Server/routes/adresses |  |
|  | Delete handyman's skill | axios.delete('/api/v1/json/handymanAbilities/edit/'+handmanID) |  | Server/routes/handymanAbilities |  |
| Search Handyman | Retrievehandyman(all) | axios.get('/api/v1/json/handymen/') |  | Server/routes/handymen |  |
|  | Retrievehandyman | axios.get('/api/v1/json/handymen/'+handmanID) |  | Server/routes/handymen |  |
|  | Search available handymen | axios.get('/api/v1/json/handymen/search/list/') | location, skill, requestDate | Server/routes/handymen |  |
|  | Retrieve Schdule Type | axios.get('/api/v1/json//workdays/') |  | Server/routes/workdays | refer WorkDay\_SampleData Sheet |
| Booking with a Handyman | Create booking with Handyman | axios.post('/api/v1/json/requests/') | customerID, requestDate, addressID, totalCost, requestStatus | Server/routes/requests |  |
|  |  | axios.post('/api/v1/json/requestDetails/') | requestID, serviceID, cost, handymanID, requestStatusID,  expectedWorkDate, expectedWorkTime, fixedDate | Server/routes/requestDetails | Service Id codes:  1. Electrical  2. Plumbing  3. Carpentry  4. Painter  5. Civil Construction  6. General labour. |

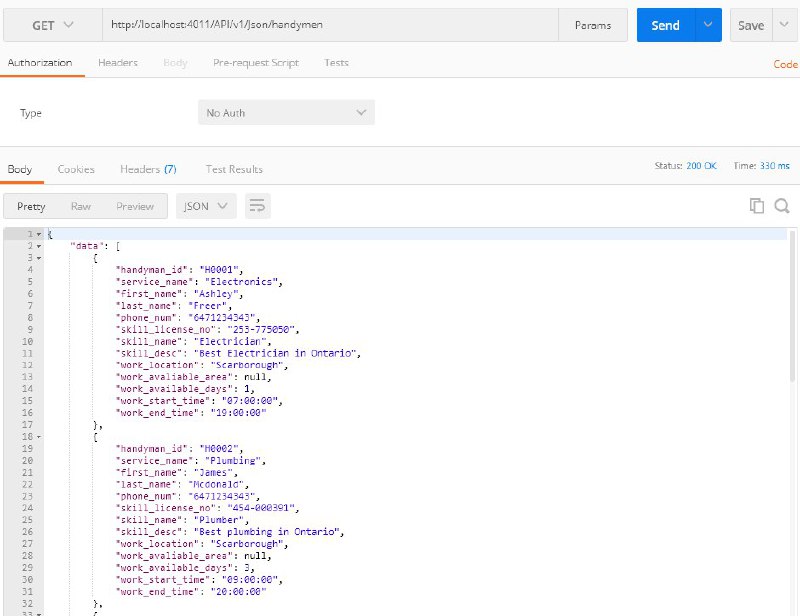
# Chapter 3

## Testing APIs

Use postman to test APIs.



Use call from front end for the end point of postman.



## Server Availability check

|  |  |
| --- | --- |
| Server Host | ISPmanager |
| Server Name: | helia |
| Database Name: | handyman |
| Server URL: | 149.202.210.182 |