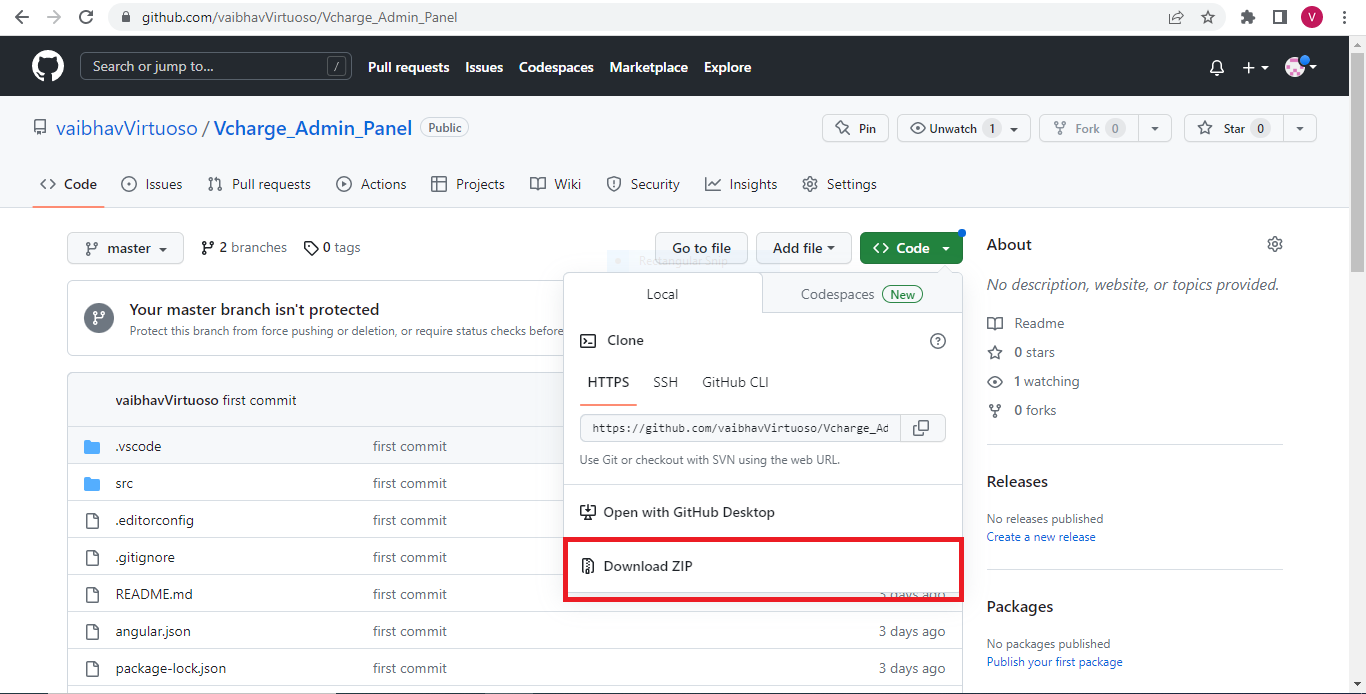
Angular Project Documentation: Admin Panel Application

**Introduction: -**

We are making website for admin, host, vendor from which they can control and view station, charger and connector details, revenue generated by the stations and chargers, etc. In this website, there will be only login page from which admin, host and vendor will be login. For admin all the feature and functionality will be accessible but for vendor and host some of the feature and functionality will be hidden for them.

**How to download the project:-**

Step 1:- Go to this link <https://github.com/vaibhavVirtuoso/Vcharge_Admin_Panel> and click on code button. Then download Zip file.



Step 2:- After downloading Vcharge\_Admin\_panel.zip file then extract the file and you can use that project now.

\*Another Way to download the same project is below (you will need to install git tool into your pc)

Step 1:- copy this command

“git clone <https://github.com/vaibhavVirtuoso/Vcharge_Admin_Panel.git>”

Step 2:- open command window of folder location where you want to download the project and simply paste the command there & run it. All the files of this project will be downloaded.

**How to Use this project:-**

Step 1:- open the project in any IDE (preferred VS Code).

Step 2:- run this command in project terminal

npm install (for installing node module)

Step 3:- then install angular material with deep purple / Amber theme only

ng add @angular/material

choose deep purple/Amber theme

Step 4:- open terminal and run the command

ng serve -o

this will open the project into your browser.

**File Structure: -**

**Components created: -**

All the components are created into app folder (Admindashboard/src/app/)

For more detailed view refer to this link:-

<https://whimsical.com/admindashboard-component-flowchart-LaRWQCp26W3n3vFKhCZi38>

Bank-details

Body

Booking

Dashboard

Downloads

Earning

Header

My-station 🡪 add-station, charging-station,station-control-access 🡪 charger-setting, connector 🡪connector-setting.

Payments

Profile

Settings 🡪 notification-setting, payment-setting, security-setting (sub Components)

Settlements

Sidenav 🡪 Sublevel-menu

Support-status

**Service created: -**

There is folder for services inside app/service where all the services are created.

myStation

charger

connector

**Development of application:-**

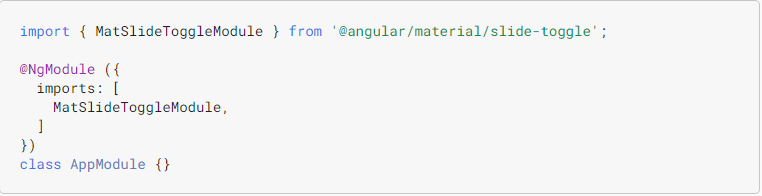
**Angular material:-**

Installed angular material for using predesigned components into project.

Step 1:- write this command for installation material into project

ng add @angular/material

Step 2:- then import the particular component which you are going to use into your app.module.ts file



Step 3:- use that particular component into your app.component.html or any other components html file



**App component page:-**

**Properties:-**

1. isSideNavCollapsed:-

* Type :- Boolean
* Used for collapsing the sidenav bar

1. screenWidth:-

* Type:- number
* Assigned zero value

**Methods:-**

1. onToggleSideNav:-

* return:- void
* parameters:-

1. data:-

* type = SideNavToggle(interface)
* it is used to listen the changes to HTML elment (input)
* this method is used to assign value to screenwidth and isSideNavCollapsed when click on sidenav bar.

**Body:-** (D:\Project\Admindashboard\src\app\body\body.component.ts)

**Input:-**

1. collapsed:-

* Type :- Boolean
* Used for storing the true or false value of sidenav is it collapsed or expanded.

1. screenWidth:-

* Type:- number
* Assigned zero value intially

**Methods:-**

1. getBodyClass:-

* return:- string
* this method is used to styleClass according to collapsed and screenWidth.

**Side-nav page:-**

**constructor:-**

parameter:-

1. route :- type = Router , used for navigation

**Properties:-**

1. collapsed:-

* Type :- boolean
* Assigned false value to this property in starting(in starting we can see navbar)

1. screenWidth:-

* Type:- number
* Assigned zero value

1. navData:-

* Type :- navbaarData(interface)
* Used for displaying Matpaginator

1. multiple:-

* Type:- boolean
* Assigned false value to this property in starting

**Methods:-**

1. toggleCollapse:-

* return:- void
* this function is responsible for collapse and expansion of sidenav bar

1. closeSidenav:-

* return:- void
* used for closing the sidenav bar

3. handleClick:-

* return:- void
* parameters:-

1. **item**:-

* type:- INavbarData(interface)
* used to take the **stationId** of staion on which admin will click
* used for handling click of sidenav menu list when we click on compenent which are have sublevel menu list.

1. getActiveClass:-

* return:- string
* parameter:-

1. **data**:-

* type:- INavbarData(interface)
* this method is used to add “active” class to particular sidenav bar list item on which we had clicked.

1. openAddStationDailog:-

* return:- void
* used for opening dialogbox of Add station form.

1. ngOnInit:-

* return:- void
* reduce the screenWidth of body component

Ouputs:-

1. onToggleSideNav:-

* Type:- EventEmitter<SideNavToggle>
* Used for listens to a named event, fires a callback, then emits that event with a value.

**Sublevel component page:-** (src/app/sidenav/sublevel-menu.component.ts)

**constructor:-**

parameter:-

1. route :- type = Router , used for navigation

**Inputs:-**

1. animating:-

* Type :- Boolean
* Used for adding animation.

1. data:-

* Type:- INavbarData()
* This contain various field like routeLink,icon,label and items.

1. multiple:-

* Type:- Boolean
* Initially its value is false
* Used for showing the active sidenav item when we click on sublevel items.

1. collapsed:-

* Type:- Boolean
* Initially its value is false
* Used for collapsing the sidenav bar

1. expanded:-

* Type:- Boolean | undefined
* Used for expanding the sidenav bar.

**Methods:-**

1. getActiveClass:-

* return:- string
* parameters:-

1. item:-

* type = INavbarData(interface)
* it contains the data of sublevel item
* it is used adding class="active-sublevel" CSS to sub-component which is active

2 handleClick:-

* return:- void
* parameters:-

1. item:-

* type = any
* it contains the data of sublevel item
* it is used handing the click for sublevel menu list.

**Dashboard page:- (**src/app/dashbord/dashboard.component.ts**)**

**Properties:-**

1. selectedOption:-

- datatype = string

- used for ngIf purpose to display the calendar if clicked on daily option and month dropdown when clicked on month option.

**myStation page:- (**src/app/my-station/my-station.component.ts**)**

**constructor:-**

parameter:-

1. myStation :- type = MystationService , used for using Mystation service
2. route :- type = Router , used for navigation
3. dialog:- type = MatDailog , used for opening Dialogbox for add station

**Properties:-**

1. DataSource:-

* Type :- MatTableDataSource<any>
* Used for storing the data which will come from services/api and this data will be used in html.

1. displayedColumns:-

* Type:- string[]
* Used for displaying columns in html

1. Paginator:-

* Type :- Matpaginator
* Used for displaying Matpaginator

1. Sort:-

* Type:- Matsort
* used for sorting the column data of each field.

1. totalNoOfChargers:-

* Type:- number
* Used for keeping the count of total no of charger

1. availableChargers:-

* Type:- number
* Used for keeping the count of total no of available charger

1. inuseChargers:-

* Type:- number
* Used for keeping the count of total no of inuse charger

1. outOffOrderChargers:-

* Type:- number
* Used for keeping the count of total no of outoff order charger

**Methods:-**

1. getStationInfo:-

* return:- void
* this function will be calling the **getMyStation()** function which is defined in **myStaion** service. **getMyStation()** will return the all station list to this function and which will store in **res**. Now **res** will assign the list of station to **dataSource.**

1. onToggleChange:-

* return:- void
* parameters:-

1. status:-

* type:- any
* used for taking the status of station(active or in active)

1. id:-

* type:- any
* used for taking the **stationId**
* used for changing the value of **stationStatus** (from active to inactive and vice versa) in database when we change the status in our website buy using toggle button.

3. onClickedSetting:-

* return:- void
* parameters:-

1. **stationId**:-

* type:- any
* used to take the **stationId** of staion on which admin will click
* used for redirecting to control access page of particular station on which admin has clicked.

1. openChargerList:-

* return:- void
* parameter:-

1. **id**:-

* type:- any
* take the station id
* when clicked on the station name then it will direct to charger list where all the list of charger is present.

1. openAddStationDailog:-

* return:- void
* used for opening dialogbox of Add station form.

1. ngOnInit:-

* return:- void
* called the **getStationInfo()** function

**add-station page:- (**src/app/my-station/add-station/add-station.component.ts**)**

**constructor:-**

parameter:-

1. formBuilder :- type = FormBuilder
2. myStation :- type = MystationService , used for using Mystation service
3. dialogRef :- type = MatDailogRef<> , used for referring Dialogbox of add station
4. @Inject(MAT\_DAILOG\_DATA) public data:- type = any, used to inject the value into constructor. When the component is constructed, Angular will set the value of this property to the data that was passed into the dialog.
5. snackBar:- type = MatSnackBar, used for using Mat sanckbar feature.

**Properties:-**

1. addStation:-

* Type :- FormGroup

1. amenities:-

* Type:- Array<any>
* Used for storing key value pair for amenities.

1. powerStation:-

* Type :- array
* Used for storing the dropdown data of power station standard

1. parkingType:-

* Type:- array
* used for storing the dropdown data of station type

1. data (public):-

* Type:- any
* Injected in add-staion component

**Methods:-**

1. onCheckboxChange:-

* return:- void
* parameters:-

1. event:-

* type = any
* it is used to listen the changes to HTML elment (input)
* this method is used for pushing the value of key when we clicked on checkbox

2. openSnackBar:-

* return:- void
* parameters:-

1. message:-

* type:- any
* the message we want to display will be store in this property

1. action:-

* type:- any
* action we want to perform like close
* used for displaying the message after certain action are performed like for example when we submit then it will help us to display “form submitted successfully”.

3. onFormSubmit:-

* return:- void
* this method is used for calling the addStationList() method(help to post the data into database) which is defined in myStation service.

**Charging Station page:- (**src/app/my-station/charging-station/charging-station.component.ts**)**

**constructor:-**

parameter:-

1. route :- type = Router , used for navigation
2. myStation :- type = MystationService , used for using Mystation service
3. charger :- type = CharagerService , used for using Charger station services.
4. activeRoute:- type = ActivatedRoute, used for using to traverse the RouterState tree and extract information from nodes.

**Properties:-**

1. stationId:-

* Type :- any
* Used for storing station Id which will be getting from activeRoute params.

1. stationName:-

* Type:- any
* Used for storing the station name.

1. dataSource:-

* Type :- MatTableDataSource<any>
* Used for storing the data which will come from services/api and this data will be used in html.

1. displayedColumns:-

* Type:- string[]
* Used for displaying columns in table.

1. totalConnectors:-

* Type :- number
* Used for storing the count of total no of charger present in that particular charger.

1. activeConnectors:-

* Type :- number
* Used for storing the count of total no of active charger present in that particular charger.

1. inactiveConnectors:-

* Type :- number
* Used for storing the count of total no of inactive charger present in that particular charger.

1. Paginator:-

* Type :- Matpaginator
* Used for displaying Matpaginator

1. Sort:-

* Type:- Matsort
* used for sorting the column data of each field.

**Methods:-**

1. getChargerListById:-

* return:- void
* parameters:-

1. id:-

* type = any
* this id is station Id on which admin will click
* this method is used for getting all the chargers which present in that particular station on which admin will click by passing the stationId as an arguments.

2. openChargerSetting:-

* return:- void
* parameters:-

1. data:-

* type:- any
* all the data of particular charger is stored onn which admin has click is stored.
* used for route navigation to charger-setting when clicked on setting button of particular charger.
* And also this function transfer the chargerId in the url for fetch the data of that chargerId in another component.

3. onConnector-

* return:- void
* parameters:-

1. chargerId:-

* type:- any
* variable in which chargerId is stored by passing it as arguments.
* this method is used redirect to connector list using stationId and chargerId.

**Station-control-access page:- (**src/app/my-station/station-control-access/station-control-access.component.ts**)**

**constructor:-**

parameter:-

1. formBuilder :- type = FormBuilder
2. activeRoute:- type = ActivatedRoute, used for using to traverse the RouterState tree and extract information from nodes.
3. myStation :- type = MystationService , used for using Mystation service.

**Properties:-**

1. stationId:-

* Type :- any
* Used for storing station Id which will be taken from paramMap.

1. myStationData:-

* Type:- any
* Used for storing station data of specific station Id.

1. apiAmenities:-

* Type :- array []
* Used for Amenities data which will be coming from api.

**Methods:-**

1. getMyStationUsingId:-

* return:- void
* parameters:-

1. id:-

* type = any
* stationId is passed as arguments.
* this method is used for getting the station details using stationId by called getStationById() which is in station Service.

**Charger-setting page:- (**src/app/my-station/add-station/add-station.component.ts**)**

**constructor:-**

parameter:-

1. formBuilder :- type = FormBuilder
2. charger :- type = CharagerService , used for using Charger station services.
3. activeRoute:- type = ActivatedRoute, used for using to traverse the RouterState tree and extract information from nodes.

**Properties:-**

1. chargerSetting:-

* Type :- any

1. chargerId:-

* Type:- any
* Used for storing chargerId which will be getting from paramMap

1. isChecked:-

* Type :- boolean
* Used for enabling the update option in the charger form

1. chargerFormData:-

* Type:- any
* used for storing charger data which will be coming from the charger service as result.

**Methods:-**

1. chargerDetailsUsingId:-

* return:- void
* parameters:-

1. id:-

* type = any
* chargerId is passed as arguments.
* this method is used for getting the charger details using chargerId by called getChargerId() which is in charger Service.

2. closeUpdateCharger:-

* return:- void
* applied this method on cancel button of charger setting for disabling the updating option.

3. updateChargerForm:-

* return:- void
* this method is used for calling updateCharger() method in the chargerService which will put the data into database.

**Connectors page:- (**src/app/my-station/charging-station/connectors/connectors.component.ts**)**

**constructor:-**

parameter:-

1. route :- type = Router , used for navigation
2. connector:- type = ConnectorService , used for using connector service
3. activeRoute:- type = ActivatedRoute, used for using to traverse the RouterState tree and extract information from nodes.
4. dialog:- type = MatDialog, used for dialog box

**Properties:-**

1. stationId:-

* Type :- any
* Used for storing station Id which will be getting from activeRoute params.

1. chargerId:-

* Type:- any
* Used for storing charger Id which will be getting from activeRoute params.

1. dataSource:-

* Type :- MatTableDataSource<any>
* Used for storing the data which will come from services/api and this data will be used in html.

1. displayedColumns:-

* Type:- string[]
* Used for displaying columns in table.

1. Paginator:-

* Type :- Matpaginator
* Used for displaying Matpaginator

1. Sort:-

* Type:- Matsort
* used for sorting the column data of each field.

**Methods:-**

1. getConnectorUsingIds:-

* return:- void
* parameters:-

1. stationId:-

* type = any
* this id is station Id which is passed as the arguments

1. chargerId:-

* type = any
* this id is charger Id which is passed as the arguments
* this method is used for getting all the connector list which present in charger of specific station.
* This method called the getConnector() present in the connector service which will return res. In res all the connector data will be present.

2. onUpdateConnector:-

* return:- void
* parameters:-

1. data:-

* type:- any
* all the data of particular connector is stored.
* used for opening the dialog box which contain form for updating the connectors data.

3. openDialogBox-

* return:- void
* this method is used for dialogbox which contains add connector form.

4 openconnectorSetting-

* return:- void
* parameters:-

1. id:-

* type:- any
* this id contains the connector Id.
* Used for navigating to charger-setting and and transfer the connectorId.

**Connector-setting page:- (**src/app/my-station/charging-station/connectors/connector-setting/connector-setting.component.ts**)**

**constructor:-**

parameter:-

1. formBuilder :- type = FormBuilder
2. connector :- type = ConnectorService , used for using connector services.
3. activeRoute:- type = ActivatedRoute, used for using to traverse the RouterState tree and extract information from nodes.
4. dialogRef :- type = MatDailogRef<> , used for referring Dialogbox of add station
5. @Inject(MAT\_DAILOG\_DATA) public data:- type = any, used to inject the value into constructor. When the component is constructed, Angular will set the value of this property to the data that was passed into the dialog.

**Properties:-**

1. addConnector:-

* Type :- FormGroup

1. connectorId:-

* Type:- any
* Used for storing chargerId which will be getting from paramMap.

1. connectorType:-

* Type :- array[]
* Used for storing the default value for connector type.

1. connectorSocket:-

* Type:- array[]
* used for storing the default value of connector socket.

1. modifiedBy:-

* Type:- array[]
* used for storing the default value of modifiedBy.

**Methods:-**

1. onFormSubmit:-

* return:- void
* this method is used for updating and adding the connector into database.

**Services:-**

**myStation:**(D:\Project\Admindashboard\src\app\service\mystation.service.ts**)**

**constructor:-**

parameter:-

1. http :- type = HttpCilent

**Methods:-**

1. getMyStationList():-

* return:- any
* this method is used for get complete list of station.

1. addStationToList(data: any):-

* return:- any
* arguments:-
* data:- type = any

-used for sending the stored data which we are going to add.

* this method is used for adding new station data into DB.

1. getStationById(id: any):-

* return:- any
* arguments:-
* id:- type = any
* used for passing the stationId as an argument.
* this method is used for get the details of station using stationId.

**charger:**(D:\Project\Admindashboard\src\app\service\charger\chargers.service.ts**)**

**constructor:-**

parameter:-

1. http :- type = HttpCilent

**Methods:-**

1. getChargerAllList(stationId: any):-

* return:- any
* arguments:-
* stationId:- type = any
* used for storing the stationId
* this method is used for get the complete list of charger bu passing the stationId

1. getChargerById(chargerId: any):-

* return:- any
* arguments:-
* chargerId:- type = any
* used for storing the chargerId.
* This method is used for get the information of charger using chargerId.

**connector:**(D:\Project\Admindashboard\src\app\service\connector\connectors.service.ts**)**

**constructor:-**

parameter:-

1. http :- type = HttpCilent

**Methods:-**

1. getConnector(stationId: any,chargerId: any):-

* return:- any
* arguments:-

1. stationId:- type = any

* used for storing the stationId.

1. chargerId:- type =any

* used for storing the chargerId.
* this method is used for get get all connectors details using stationId and chargerId.

1. getConnectorById(chargerId: any):-

* return:- any
* arguments:-
* chargerId:- type = any
* used for storing the chargerId.
* This method is used for get connector information by connectorId.