**HTML:**

<div>  
 <h2>Vote for your player of the season</h2>  
  
 <h3> Enter your email id to vote</h3>  
 <input type="text" name="email" #***emailValue***>  
 <button type="submit" (click)="searchMail(***emailValue***.value)">Enter</button>  
 <ul>  
 <li *\*ngFor*="let ***player*** of playerData">  
 <img [src]="***player***.image" [alt]="***player***.name" (click)="castVote(***player***.shortName, ***emailValue***.value)" [ngClass]="getVoteClasses(***player***.shortName)">  
  
 <a href="{{***player***.link}}" target="\_blank"><h4>{{***player***.name}}</h4></a>  
 <p>{{***player***.Runs}} Runs</p>  
 <p>{{***player***.matches}} Matches</p>  
 <p>{{***player***.battingAvg}} Batting Average</p>  
 <p *\*ngIf*="***player***.bowlingAvg != '-'">{{***player***.bowlingAvg}} Bowling Average</p>  
<!-- <p>{{player.Wickets}} Wickets</p>-->  
<!-- <p>{{player.Catches}} Catches</p>-->  
 </li>  
  
 </ul>  
  
 <div class="chart-box" *\*ngIf*="voted">  
 <h2>How others voted</h2>  
 <canvas baseChart [data]="chartData" [labels]="chartLabels" [chartType]="chartType"></canvas>  
 </div>  
</div>

**CSS:**

div {  
 width: 80%;  
 margin: auto;  
 text-align: center;  
 h2{  
 font-size: 50px;  
 opacity: 0.9;  
 margin-bottom: 8px;  
 font-weight: bolder;  
  
 }  
 h3{  
 font-size: 20px;  
 opacity: 0.9;  
 margin-left: 2px;  
 margin-bottom: 8px;  
 font-weight: bolder;  
  
 }  
 ul {  
 list-style: none;  
 padding-left: 0;  
 display: flex;  
 justify-content: center;  
  
 li {  
 padding: 20px;  
  
 img {  
 width: 250px;  
 height: 250px;  
 border-radius: 50%;  
 box-shadow: 0 3px 4px 1px rgba(0, 0, 0, 0.1);  
 filter: grayscale(1);  
 border: 4px solid rgba(0, 0, 0, 0.2);  
 cursor: pointer;  
  
 &.elect {  
 border: 3px solid rgb(204, 54, 54);  
 box-shadow: 0 4px 7px 1px rgba(0, 0, 0, 0.1);  
 filter: grayscale(0);  
 cursor: default;  
 }  
  
 &.lost {  
 box-shadow: unset;  
 border: 4px solid rgba(0, 0, 0, 0.1);  
  
 &:hover {  
 filter: grayscale(1);  
 cursor: default;  
 }  
 }  
  
 &:hover {  
 filter: grayscale(0);  
 }  
 }  
  
 h4 {  
 font-size: 40px;  
 opacity: 0.9;  
 margin-bottom: 8px;  
 font-weight: bolder;  
 }  
  
 p {  
 font-size: 24px;  
 opacity: 0.6;  
 font-weight: bold;  
 margin: 4px 0;  
 }  
 }  
 }  
}  
  
.chart-box{  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 h2 {  
 font-size: 40px;  
 opacity: 0.9;  
 margin-bottom: 8px;  
 font-weight: bolder;  
 }  
 canvas{  
 font-size: 20px;  
 opacity: 0.9;  
 margin-bottom: 8px;  
 font-weight: bolder;  
 }  
}

**APP.COMPONENT.TS:**

import { ***Component***, OnInit } from '@angular/core';  
import { HttpClient } from '@angular/common/http';  
import { PusherService } from './pusher.service';  
@Component({  
 selector: 'app-root',  
 templateUrl: './app.component.html',  
 styleUrls: ['./app.component.scss'],  
})  
export class AppComponent implements OnInit {  
 constructor(private pusher: PusherService, private http: HttpClient) {  
 }  
  
 event = 'vote';  
 vote = '';  
 voted = false;  
 votesData;  
 displayChart = false;  
 duplicate = false;  
 playerIds = [28081,253802,219889,234675,625383];  
 //playerapiData = [];https://api.trumail.io/v2/lookups/json?email=vemula1802@gmail.com  
  
 playerData = [  
 {  
 name: 'M S Dhoni',  
 pid: 28081,  
 Runs: 0,  
 Wickets: 0,  
 Catches: 0,  
 shortName: 'Dhoni',  
 image:  
 'https://smedia2.intoday.in/aajtak/images/Photo\_gallery/092019/w43243\_030420010835\_030920012325.jpg',  
 battingAvg: 0,  
 bowlingAvg: 0,  
 matches: 0,  
 link: `https://www.cricapi.com/player/28081`  
 },  
 {  
 name: 'Virat Kohli',  
 Runs: 0 ,  
 pid: 253802,  
 Wickets: 0,  
 Catches: 0,  
 shortName: 'Kohli',  
 image:  
 'https://imagevars.gulfnews.com/2019/04/29/Virat-Kohli-gestures\_16a6847fc38\_large.jpg',  
 battingAvg: 0,  
 bowlingAvg: 0,  
 matches: 0,  
 link: `https://www.cricapi.com/player/253802`  
 },  
 {  
 name: 'David Warner',  
 Runs: 0,  
 pid: 219889,  
 Wickets: 0,  
 Catches: 0,  
 shortName: 'Warner',  
 image:  
 'https://engcric.b-cdn.net/wp-content/uploads/2019/04/GAZI\_1576.jpg',  
 battingAvg: 0,  
 bowlingAvg: 0,  
 matches: 0,  
 link: `https://www.cricapi.com/player/219889`  
 },  
 {  
 name: 'Ravindra Jadeja',  
 Runs: 0,  
 pid: 234675,  
 Wickets: 0,  
 Catches: 0,  
 shortName: 'Jadeja',  
 image:  
 'https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRwkILg1kuFbycP5dlDc1wZIic1Rd2PiAIq3QxLKMD1yBNjwyTi&usqp=CAU',  
 battingAvg: 0,  
 bowlingAvg: 0,  
 matches: 0,  
 link: `https://www.cricapi.com/player/234675`  
 },  
 {  
 name: "Jasprit Bumrah",  
 Runs: 0,  
 pid: 625383,  
 Wickets: 0,  
 Catches: 0,  
 shortName: 'Bumrah',  
 image:  
 'https://www.cricapi.com/playerpic/625383.jpg',  
 battingAvg: 0,  
 bowlingAvg: 0,  
 matches: 0,  
 link: `https://www.cricapi.com/player/625383`  
 },  
 ];  
 voteCount = {  
 Dhoni: 0,  
 Kohli: 0,  
 Warner: 0,  
 Jadeja: 0,  
 Bumrah: 0,  
 };  
 check: boolean = false;  
  
 // chartLabels: string[] = Object.keys(this.voteCount);  
 // chartData: number[] = Object.values(this.voteCount);  
 chartLabels: string[];  
 chartData: number[];  
 chartType = 'pie';  
 clickEnter: boolean = false;  
  
 getVotes() {  
 this.http.get('http://localhost:4000/getVotes')  
 .subscribe(res => {  
 ***console***.log('Response for test vote', res);  
 this.votesData = res;  
 // this.displayCharts();  
 });  
 //https://cricapi.com/api/playerStats?pid=253802&apikey=wccgJav4W5SReVtv9ZXJIShzlPR2  
 // this.http.get('https://cricapi.com/api/playerStats?pid=253802&apikey=wccgJav4W5SReVtv9ZXJIShzlPR2')  
 // .subscribe(res => {  
 // console.log('player details', res);  
 // })  
  
 }  
  
 displayCharts() {  
 this.getVotes();  
 // this.voteCount = {  
 // Dhoni: 0,  
 // Kohli: 0,  
 // Warner: 0,  
 // Jadeja: 0,  
 // Bumrah: 0,  
 // };  
 this.votesData.forEach(vote => {  
 this.voteCount[vote.playerVoted] += 1;  
 })  
 ***console***.log('this.vote@@@@@@@@@@@@',this.vote);  
 // this.voteCount[this.vote] +=1;  
 this.chartLabels = ***Object***.keys(this.voteCount);  
 this.chartData = ***Object***.values(this.voteCount);  
  
 ***console***.log('chart data voute count@@@@@@', this.voteCount);  
  
 this.displayChart = true;  
  
 }  
  
 searchMail(emailid) {  
 this.clickEnter = true;  
 if(this.ValidateEmail(emailid)){  
  
 this.check = false;  
 this.http.get('https://api.emailverifyapi.com/v3/lookups/json?key=7FEFF99F9A75E168&email='+emailid)  
 .subscribe((res: any) =>{  
 ***console***.log(' REsult of email validation', res);  
 // if(res){  
 if(!res.deliverable) {  
 this.check = true;  
 this.clickEnter = false;  
 alert("Invalid Email ID");  
 } else {  
 this.check = false;  
 this.clickEnter = true;  
 // if(!check) {  
 // this.http.post('http://localhost:4000/testVote',body).subscribe(res => console.log('Response for test vote', res));  
 // this.http  
 // .post('http://localhost:4000/vote', { player })  
 // .subscribe((res: any) => {  
 // this.vote = res.player;  
 // this.voted = true;  
 // });  
 // this.getVotes();  
 // }  
 }  
 // }  
 })  
  
 }  
 }  
  
 castVote(player,emailid) {  
 ***console***.log('check value@@@@@', this.check);  
 this.getVotes();  
 this.votesData.forEach(item => {  
 ***console***.log('Each item in votes...',item);  
 if (item.userName == emailid) {  
 this.check = true;  
 this.clickEnter = false;  
 this.duplicate = true;  
 }  
 // this.duplicate = false;  
 });  
 if(!this.check && this.clickEnter){  
 let body = {player,emailid}  
 this.http.post('http://localhost:4000/testVote',body).subscribe(res => ***console***.log('Response for test vote', res));  
 this.http  
 .post('http://localhost:4000/vote', { player })  
 .subscribe((res: any) => {  
 ***console***.log('voted player#####', res);  
 this.vote = res.player;  
 this.voted = true;  
 this.voteCount = {  
 Dhoni: 0,  
 Kohli: 0,  
 Warner: 0,  
 Jadeja: 0,  
 Bumrah: 0,  
 };  
 this.voteCount[this.vote] += 1;  
 ***console***.log('vote count........',this.voteCount);  
 this.displayCharts();  
 });  
 // this.getVotes();  
  
 } else if(this.duplicate) {  
 alert('Already casted your vote...!');  
 } else {  
 alert('Please Enter valid email to cast you vote!');  
 }  
 //console.log('entererd value', emailid);  
 }  
  
 ValidateEmail(mail)  
 {  
 if (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(mail))  
 {  
 return (true)  
 }  
 alert("You have entered an invalid email address!")  
 return (false)  
 }  
  
 getVoteClasses(player) {  
 return {  
 elect: this.voted && this.vote === player,  
 lost: this.voted && this.vote !== player,  
 };  
 }  
  
 getPlayersData() {  
 // this.playerapiData = [];  
 // this.playerIds.forEach( player => {  
 // let url = `https://cricapi.com/api/playerStats?pid=${player}&apikey=wccgJav4W5SReVtv9ZXJIShzlPR2`  
 // this.http.get(url).subscribe(  
 // playerData => {  
 // this.playerapiData.push(playerData);  
 // console.log('Players data........', this.playerapiData);  
 // })  
 // })  
 this.playerData.forEach(player =>{  
 let url = `https://cricapi.com/api/playerStats?pid=${player.pid}&apikey=wccgJav4W5SReVtv9ZXJIShzlPR2`  
 this.http.get(url).subscribe(  
 (playerData: any) => {  
 ***console***.log('api resnponse for players data...',playerData);  
 player.Runs = playerData.data.batting.T20Is.Runs;  
 player.battingAvg = playerData.data.batting.T20Is.Ave;  
 player.bowlingAvg = playerData.data.bowling.T20Is.Ave;  
 player.matches = playerData.data.batting.T20Is.Mat;  
 ***console***.log('Players data........', player.Runs);  
 })  
 })  
 }  
  
 ngOnInit() {  
 ***console***.log('entered');  
 this.getPlayersData();  
 this.getVotes();  
 const channel = this.pusher.init();  
 channel.bind('vote', ({ player }) => {  
 ***console***.log('Vote count variable...', this.voteCount);  
 // this.voteCount[player] += 1;  
 this.chartData = ***Object***.values(this.voteCount);  
 });  
 }  
}

**SERVER.JS:**

require('dotenv').config();  
const express = require('express');  
const bodyParser = require('body-parser');  
const Pusher = require('pusher');  
const mongoose = require('mongoose');  
  
const app = express();  
const port = ***process***.env.PORT || 4000;  
const pusher = new Pusher({  
 appId: '986682',  
 key: '421a81c5d52da7b3d970',  
 secret: 'c210d430673a545e2dfc',  
 cluster: 'us2',  
 encrypted: true,  
});  
  
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({ extended: false }));  
app.use((req, res, next) => {  
 res.header('Access-Control-Allow-Origin', '\*');  
 res.header(  
 'Access-Control-Allow-Headers',  
 'Origin, X-Requested-With, Content-Type, Accept'  
 );  
 next();  
});  
  
mongoose.connect('mongodb+srv://rvr68:1234567890@cluster0-c5izt.mongodb.net/test?retryWrites=true&w=majority')  
 .then(() => ***console***.log("Connected to DB Successfully"))  
 .catch(err => ***console***.log("Error in connecting to DB", err));  
  
const UserHits = new mongoose.Schema({  
 userName : {  
 type: ***String***,  
 required: true  
 },  
 playerVoted : {  
 type: ***String***,  
 required: true  
 }  
})  
  
const UserTable = mongoose.model('user-hits',UserHits);  
  
app.get('/getVotes', (req, res) => {  
 ***console***.log('test the req body...@@@@@', req.body);  
 UserTable.find().then( votes => res.send(votes));  
});  
  
app.post('/testVote', (req, res) => {  
 ***console***.log('test the req body...@@@@@', req.body);  
 const newVote = new UserTable({  
 userName: req.body.emailid,  
 playerVoted: req.body.player  
 })  
 newVote.save();  
});  
  
// pusher code  
 app.post('/vote', (req, res) => {  
 ***console***.log('test the vote req body...@@@@@', req.body);  
 const { body } = req;  
 const { player } = body;  
  
 pusher.trigger('vote-channel', 'vote', {  
 player,  
 });  
 res.json({ player });  
});  
  
app.listen(port, () => {  
 ***console***.log(`Server started on port ${port}`);  
});

**PUSHER.JS:**

import { ***Injectable*** } from '@angular/core';  
declare const Pusher: any;  
declare const promisify: any;  
  
@Injectable()  
export class PusherService {  
 constructor() {  
 var pusher = new Pusher('421a81c5d52da7b3d970', {  
 cluster: 'us2',  
 encrypted: true,  
 });  
 this.channel = pusher.subscribe('vote-channel');  
 }  
 channel;  
  
 public init() {  
 return this.channel;  
 }  
}