**PROJECT FINAL REPORT**

**PROJECT TITLE**: REAL TIME POLL SYSTEM

**PROJECT TEAM MEMBERS**:

|  |  |
| --- | --- |
| **Name** | **Class ID** |
| RAJASHEKAR REDDY VEMULA | 25 |
| KOUSHIK REDDY SAMA | 21 |
| SAI KRISHNA REDDY KATTA | 14 |

**What is the use of Real Time Polling System and how is it benefited to the public?**

A real time electronic polling system allows users cast their votes with ease without the hassle and stress of visiting a polling booth. This makes it easily accessible as it can be used by users anywhere in the world. Adding real time functionality to the application improves the user experience as votes are seen in real time.

**How a Real Time Polling System is different from a voting system?**

In the traditional voting system a user have to choose a person to vote and have to press button on the electronic voting machine (EVM) which is beside the person’s name. Where as in the real time poll system a user can vote to their favorite person by opening a web application and enter the email id for verification purpose. After the email id is verified a user can vote for their favorite player by clicking on his image and can wait to see the results in the form of a pie chart. The purpose of email id is to stop the duplication of votes by a user.

**GOALS AND OBJECTIVES OF THE PROJECT:**

The main idea of this project is to build a real time poll system, where the users can vote to their favorite cricket players through a web application and also users can see the number of votes each player received.

The goal of our project is building an efficient real time polling application using Angular, Pusher Service, MongoDB, and charts.js for data visualization. Using our application users will get to vote for their favorite cricket player in the Indian Premier League.

We’ll send our votes to the database MongoDB and with the help of MongoDB we will update polls of players in real time.

We’ll be using these tools to build our application:

* Express
* Node
* MongoDB
* Angular
* Charts
* Pusher Service

**Express**: Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

**Charts**: This tool is used for graphical representation of information and data which is called as Data Visualization.

**MongoDB**: MongoDB is a cross-platform document-oriented database program.

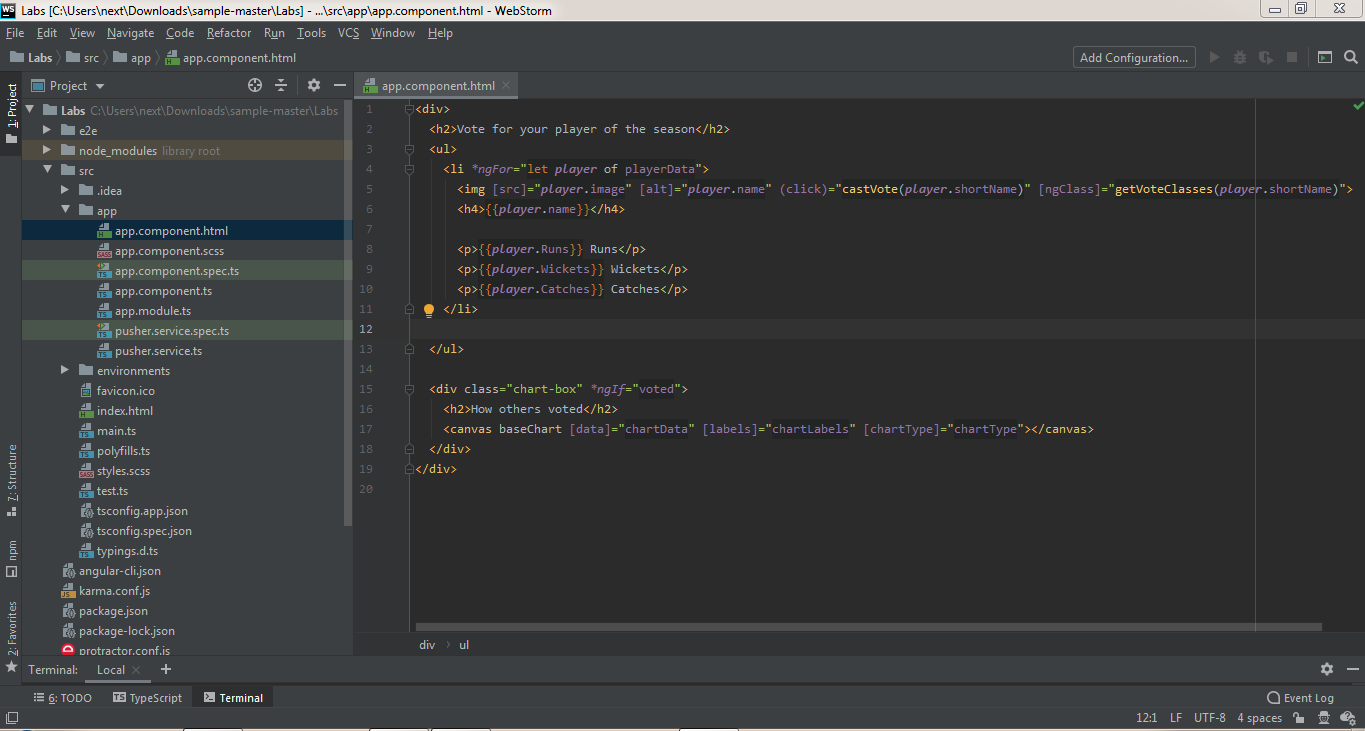
**Pusher Service**: Pusher is a hosted service that adds real-time data and functionality to web and mobile applications. Pusher sits as a real-time layer between your servers and your clients.

**Features**

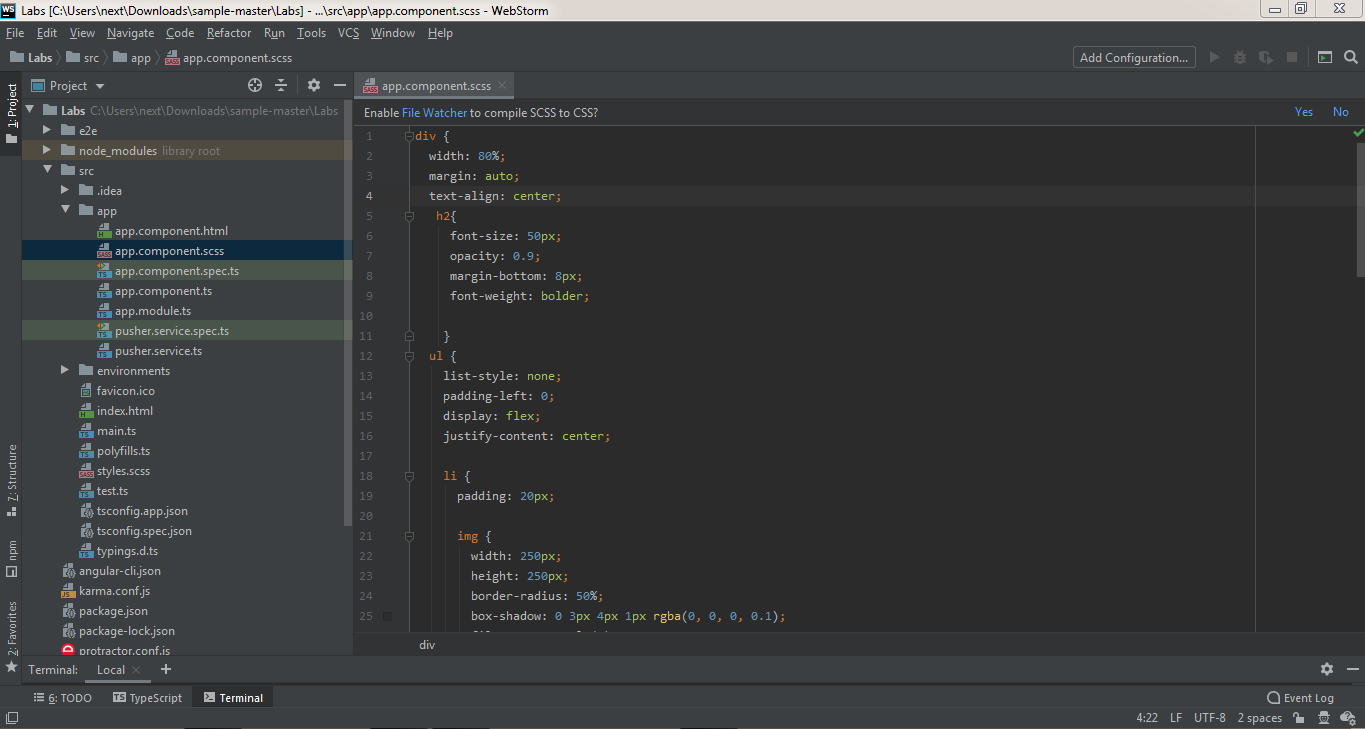
1. Email ID for Verification.
2. Selecting or voting for a favorite Player.
3. Displaying the Statistics of Each Player.
4. User can see how other users have voted for their favorite players with display of vote count in a pie chart.

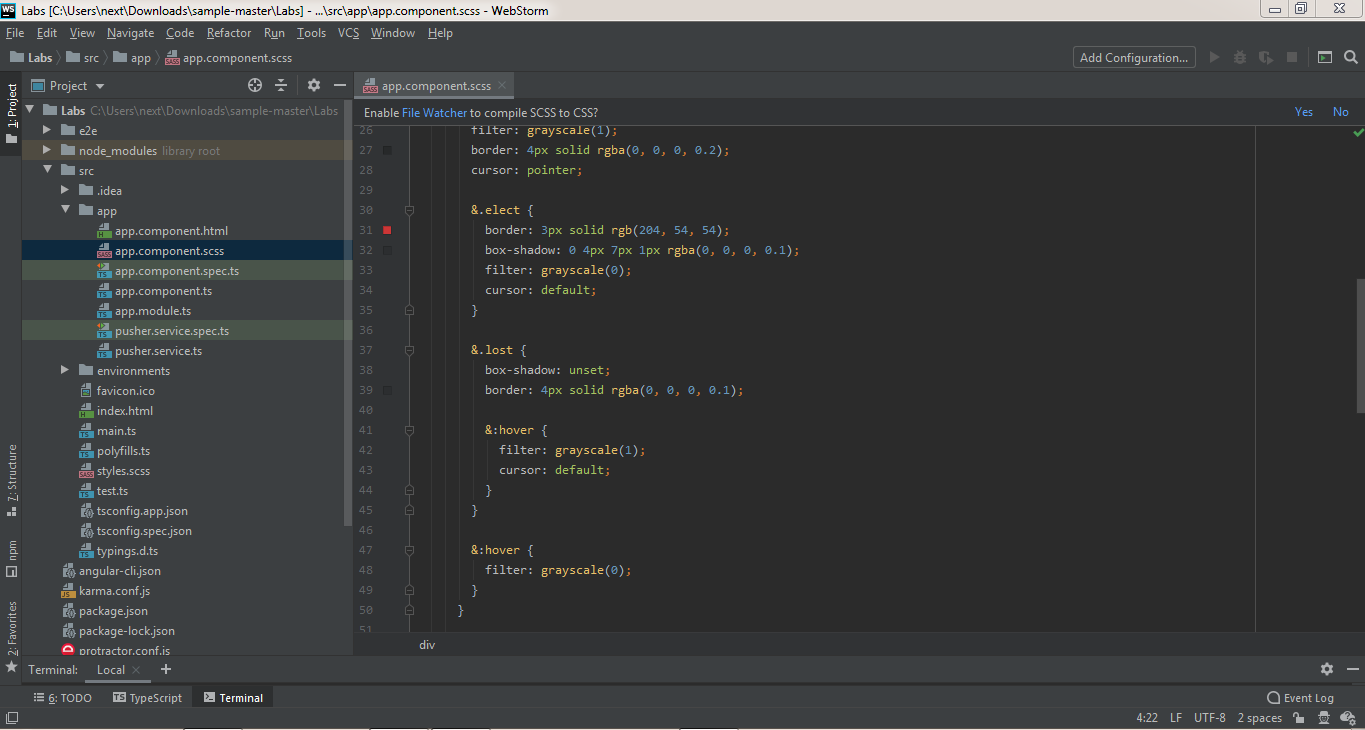
**Working Screens from Project**

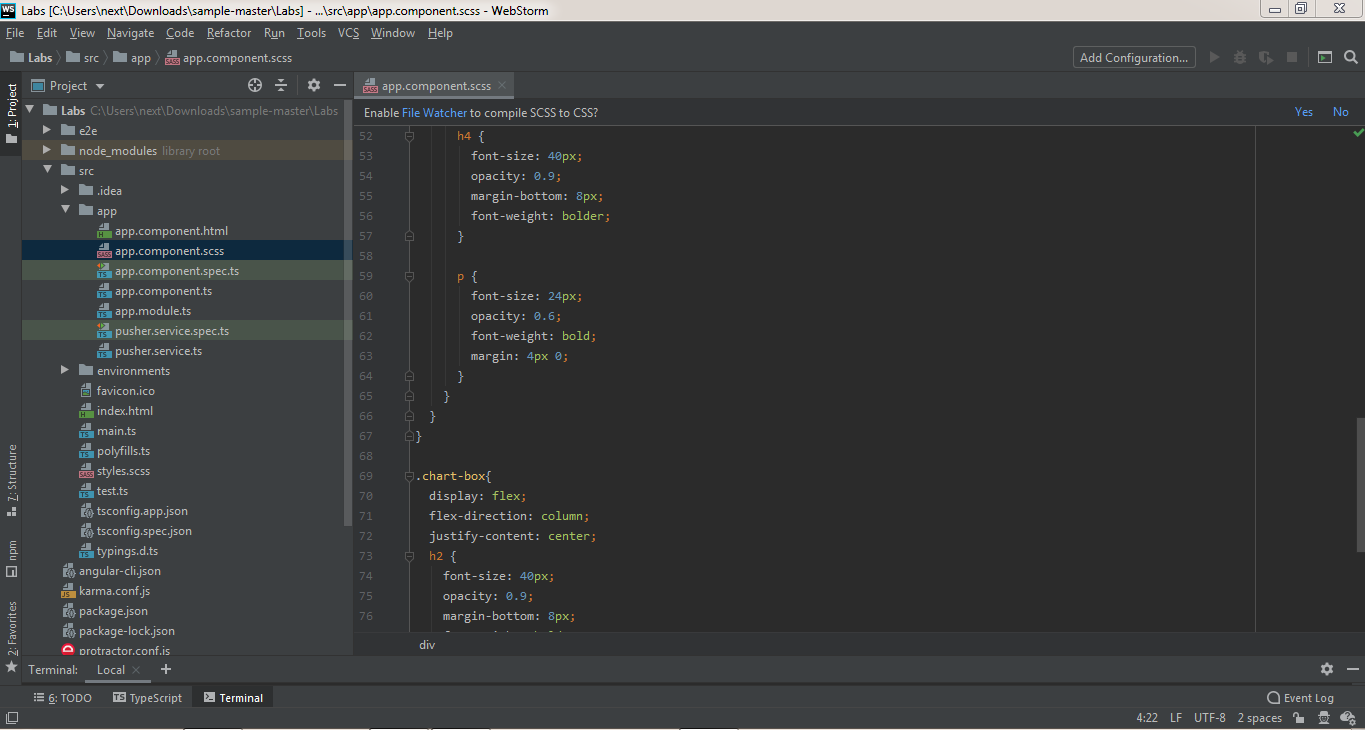
app.component.html file

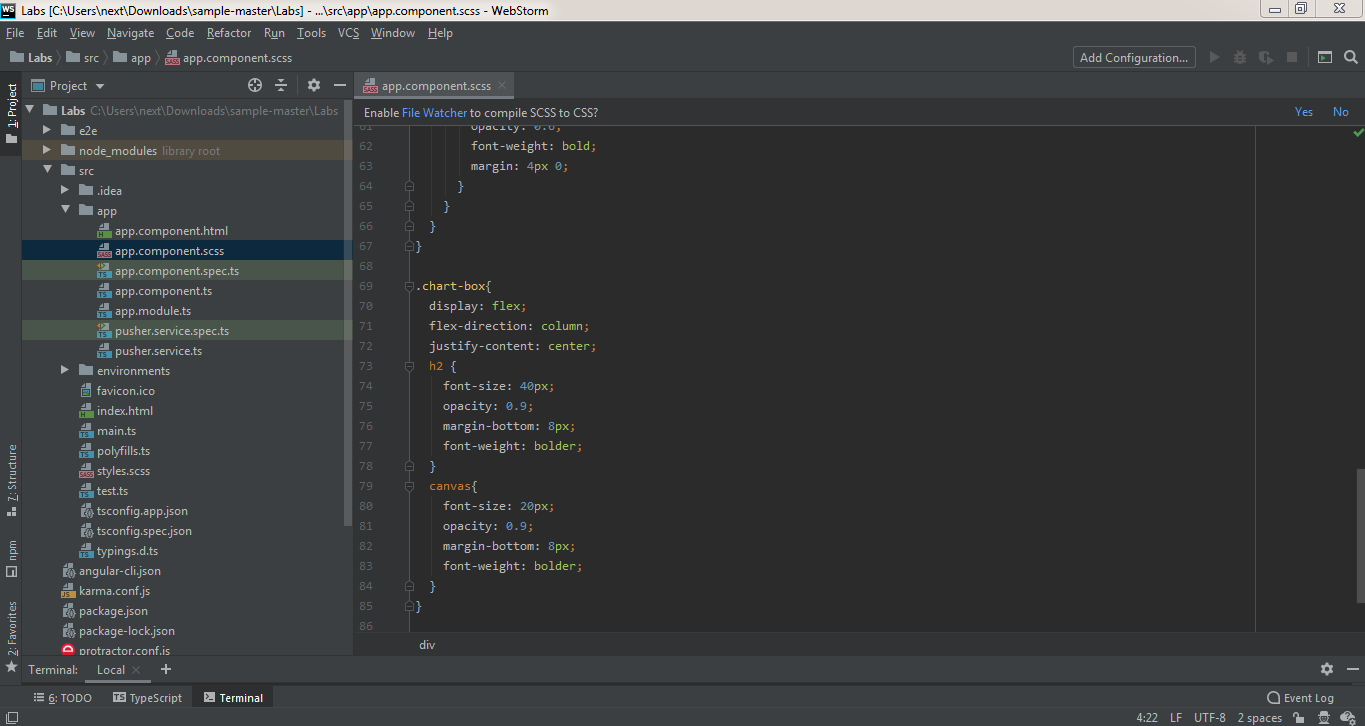


app.component.scss file

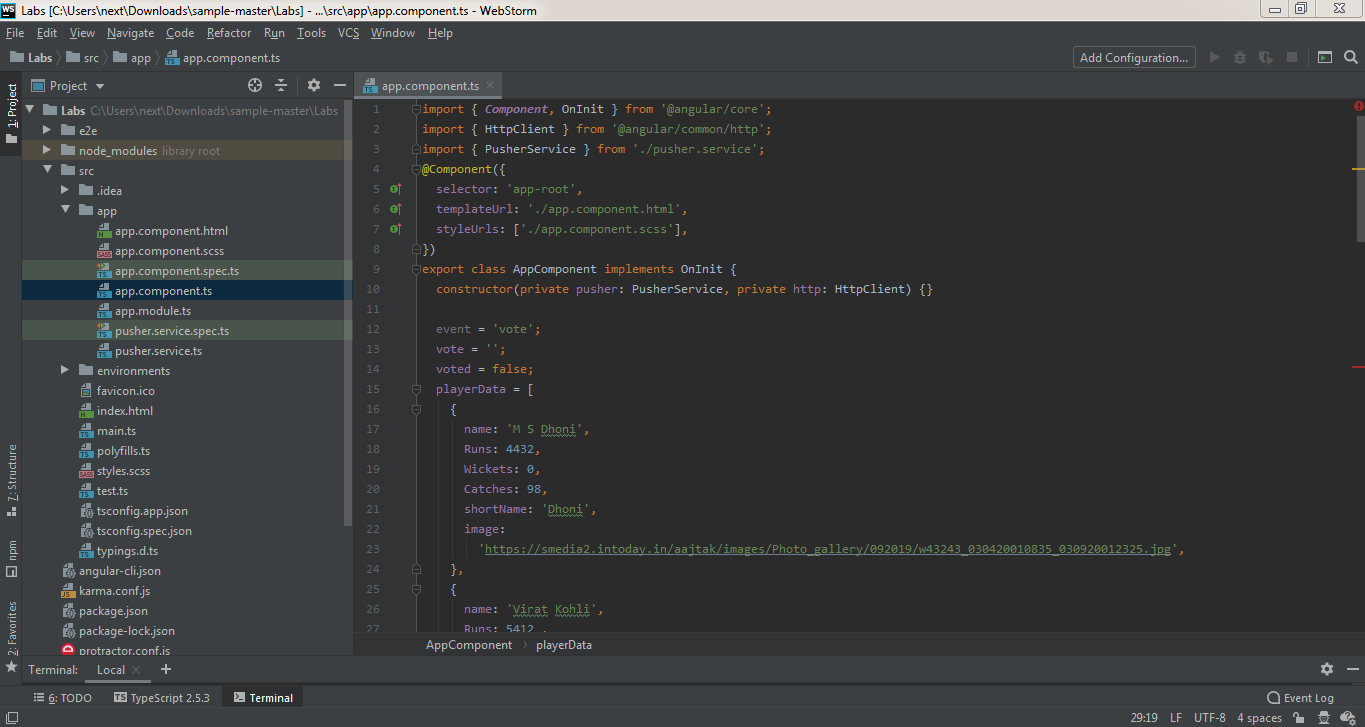


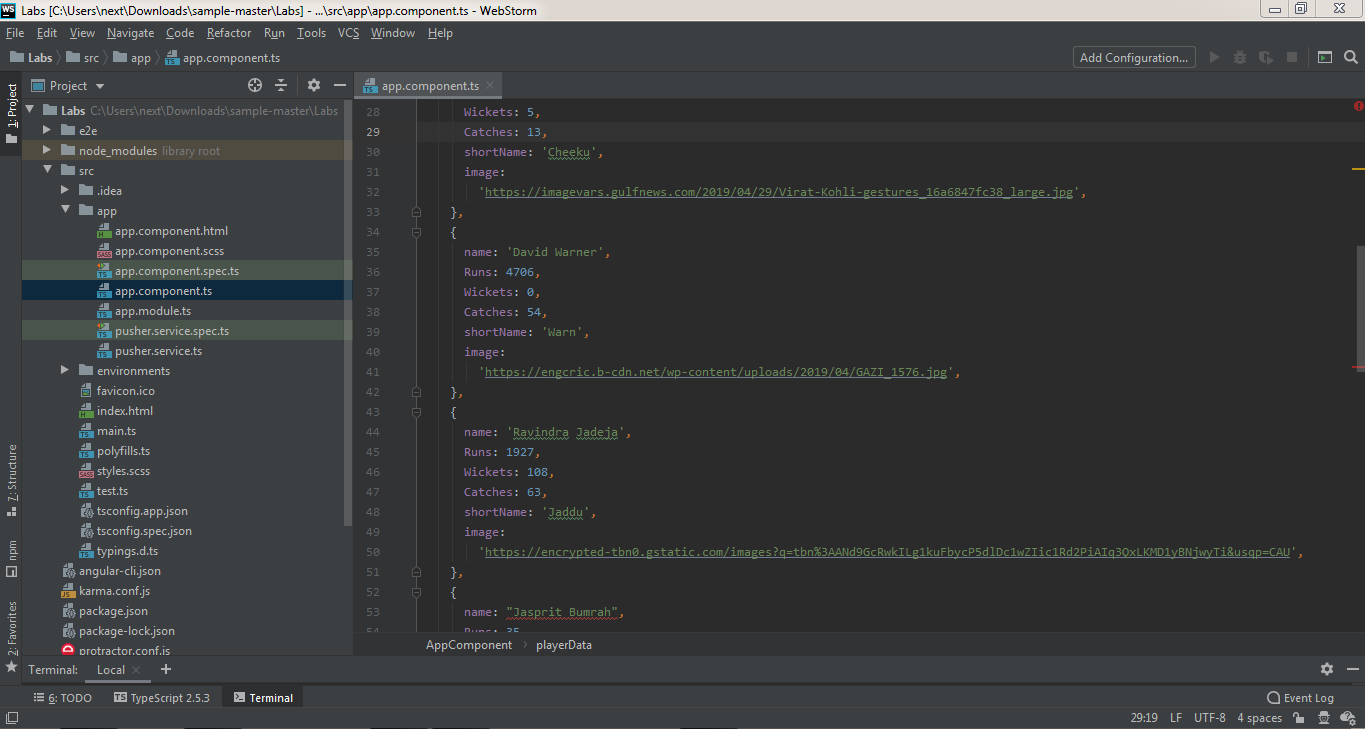


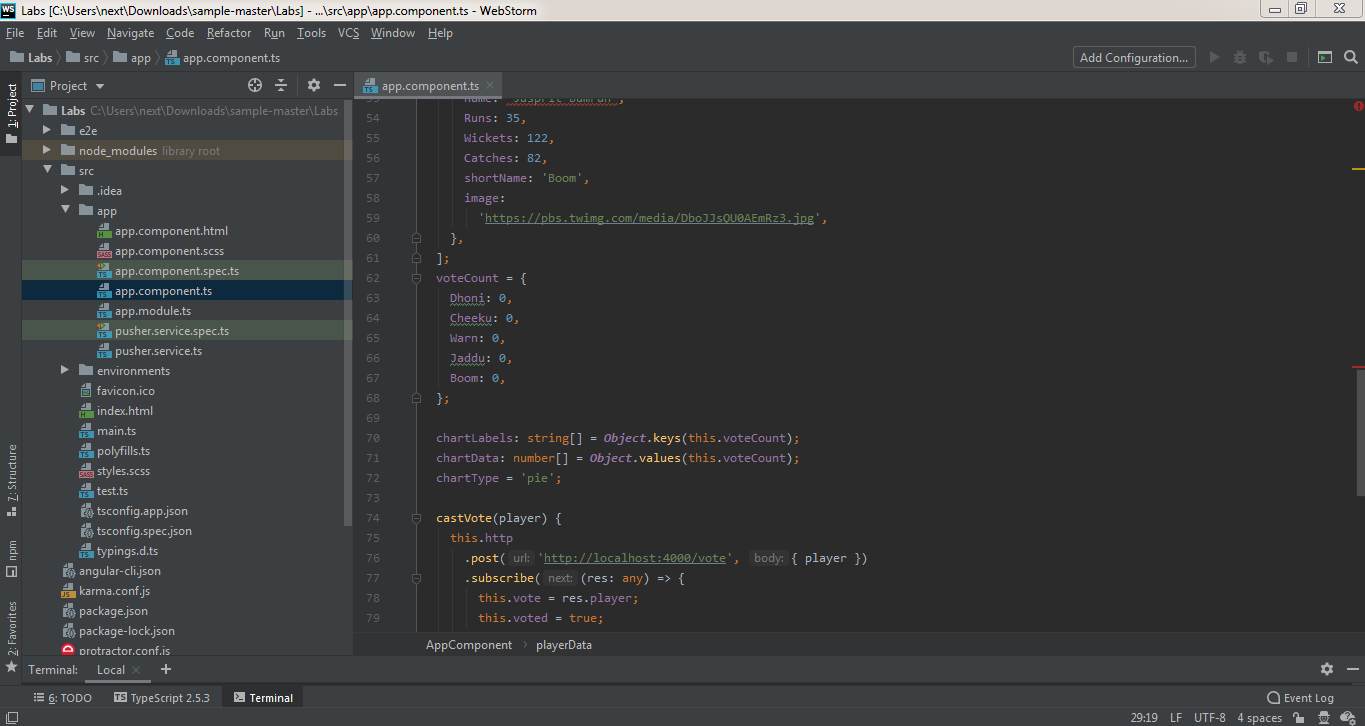


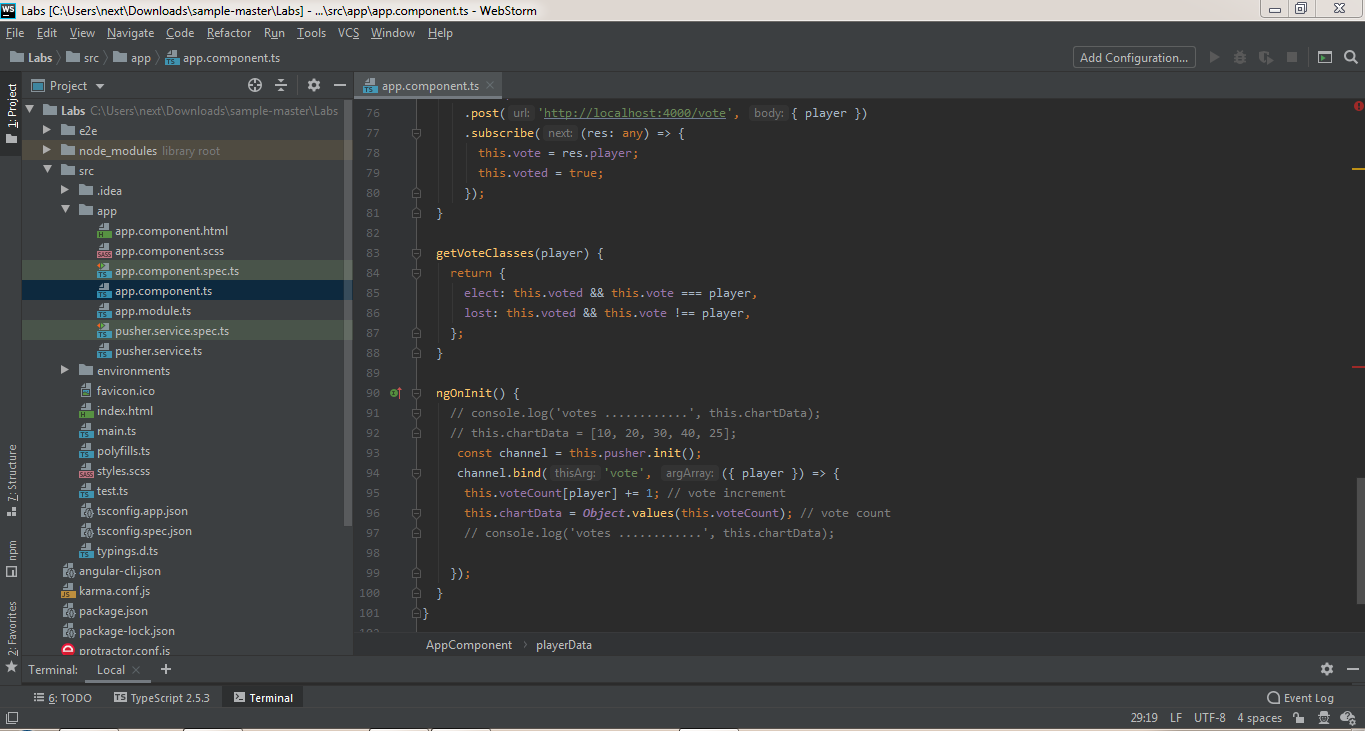


app.component.ts file

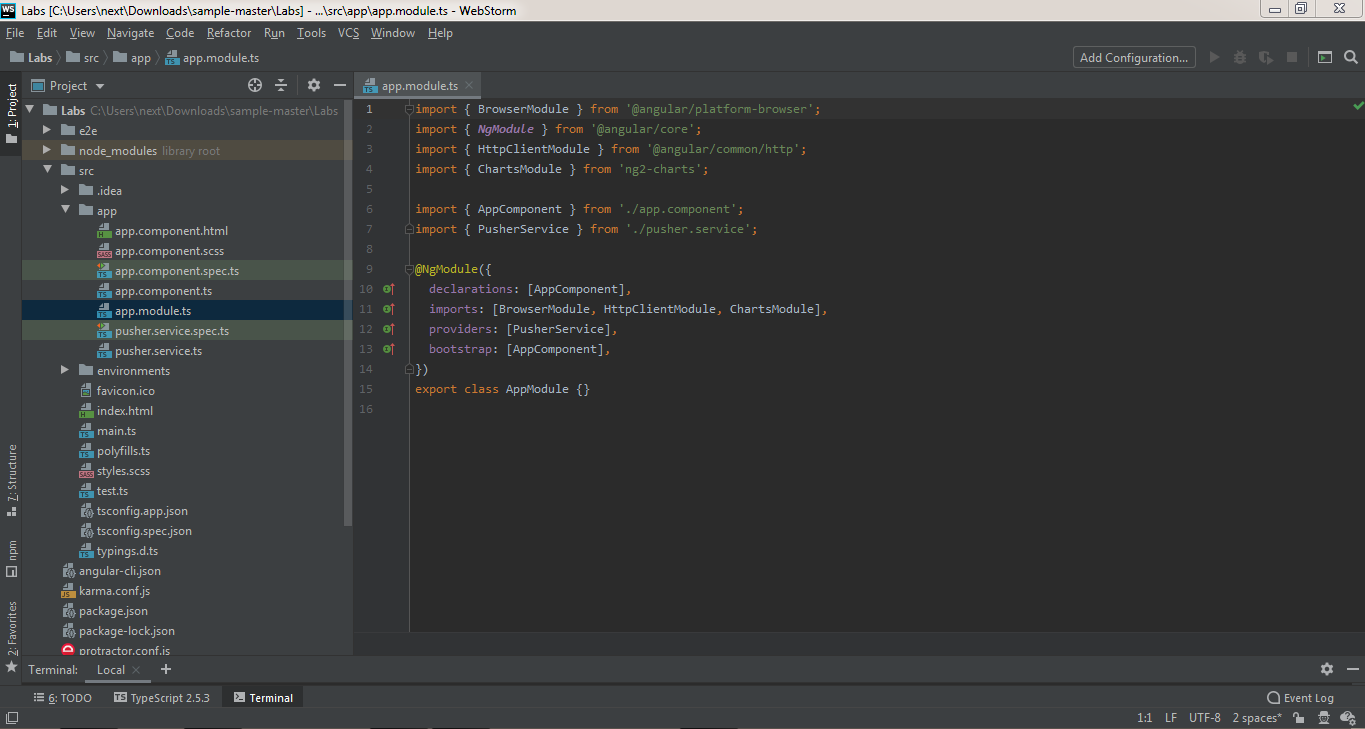




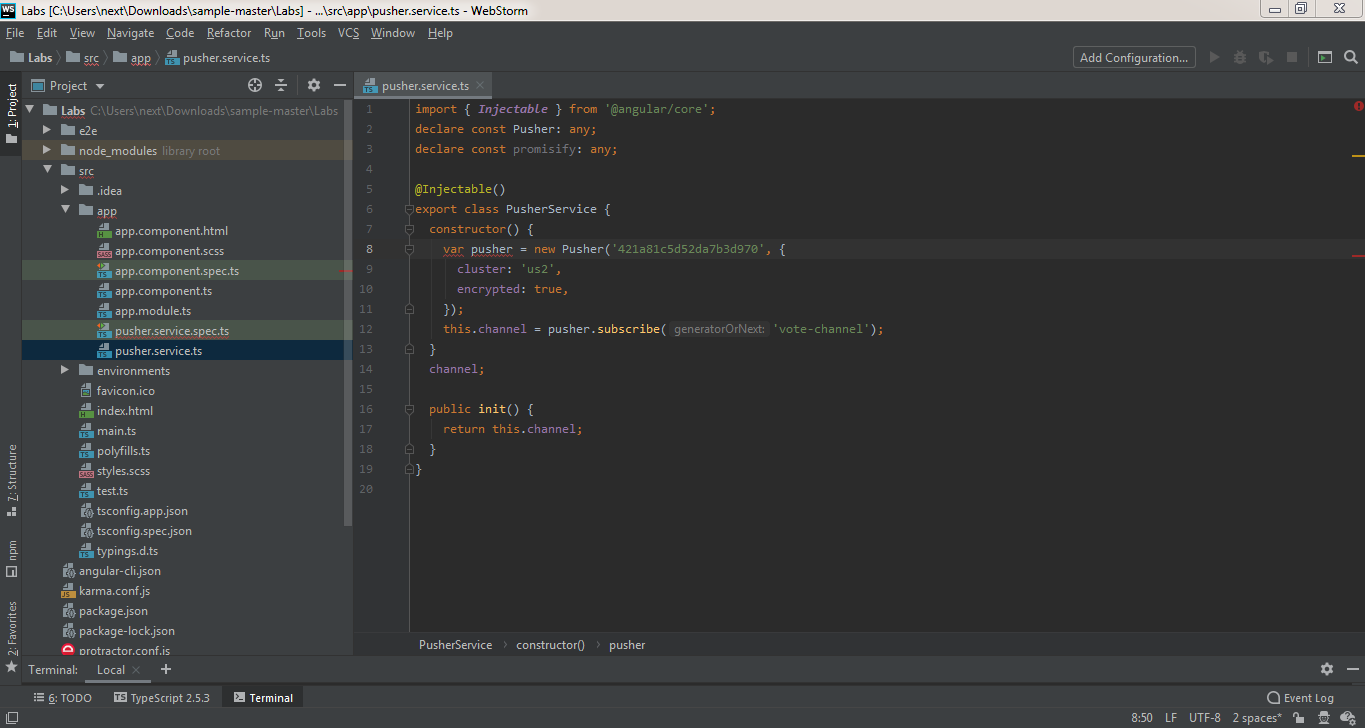


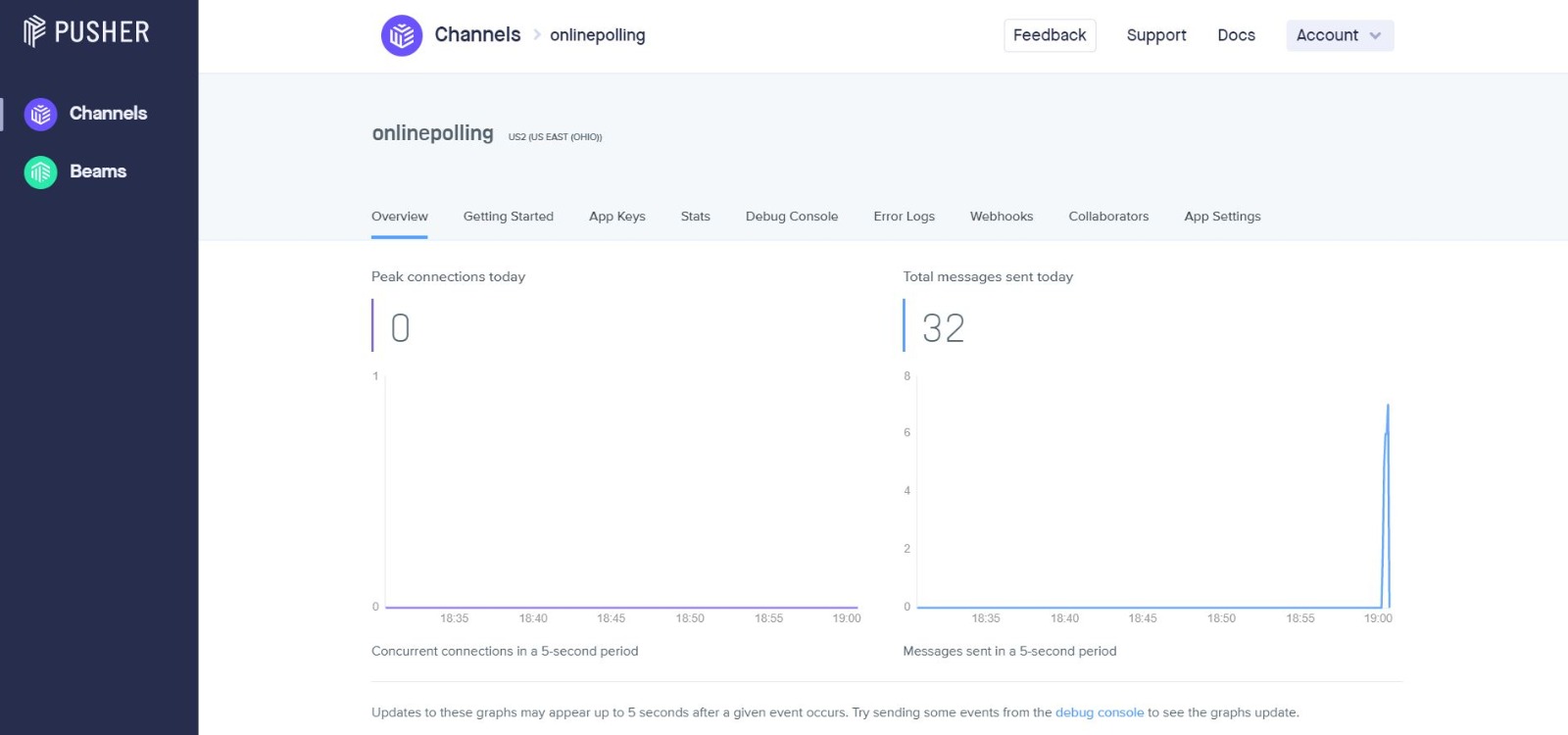


app.module.ts file

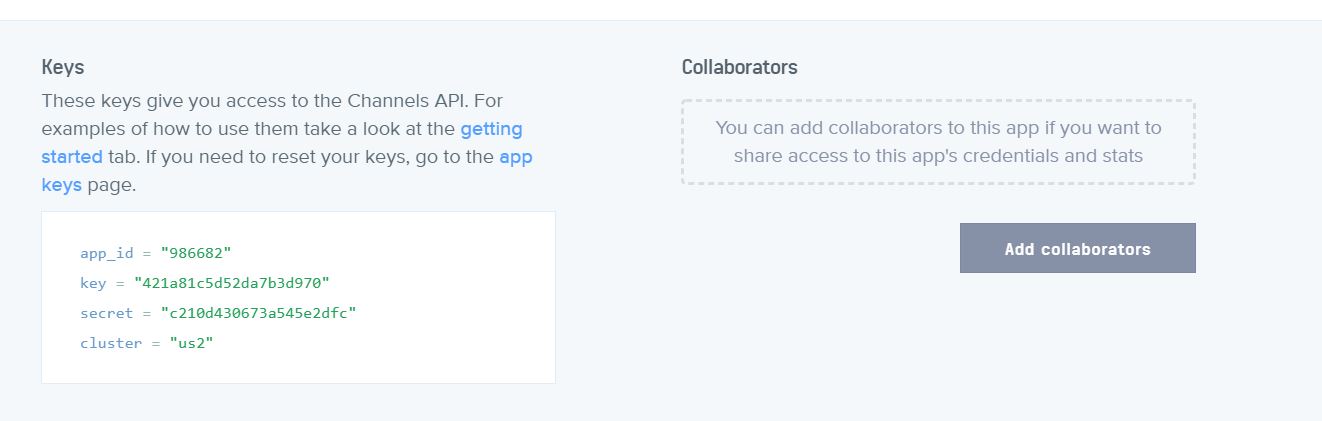


pusher.service.ts file



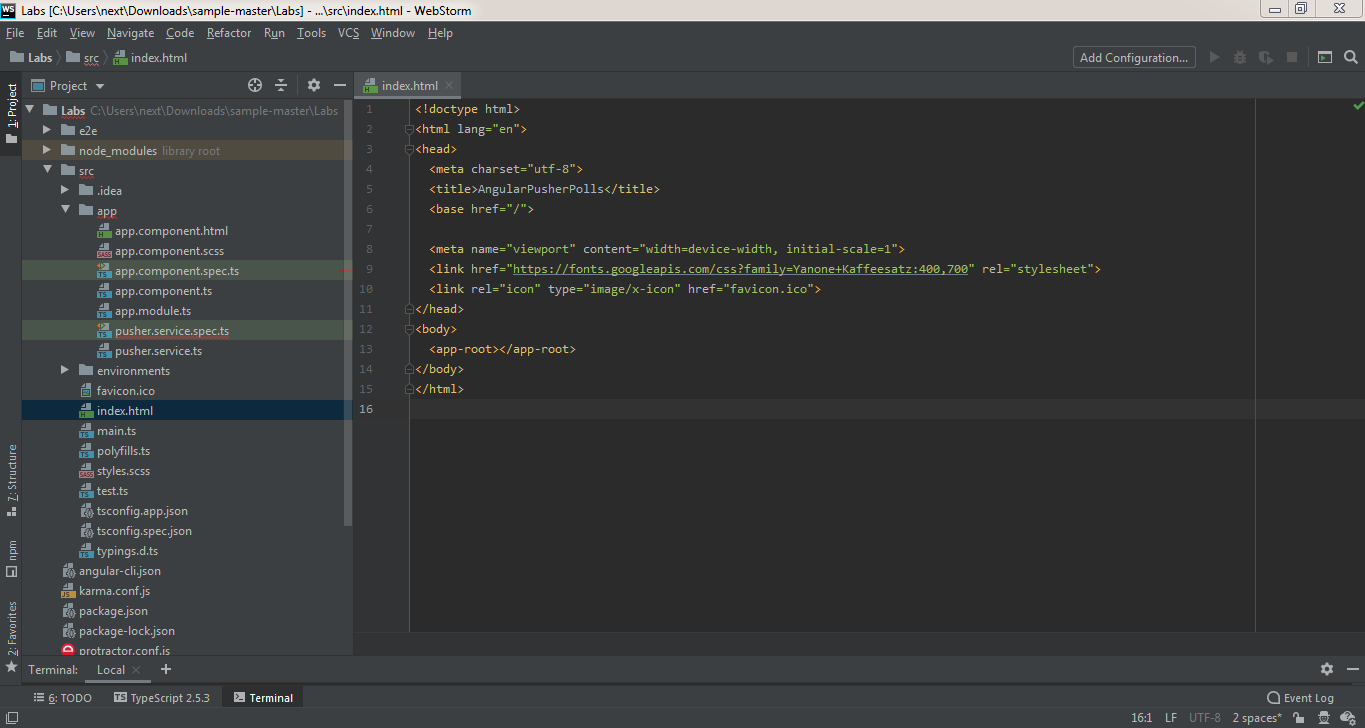


Pusher Service Dashboard

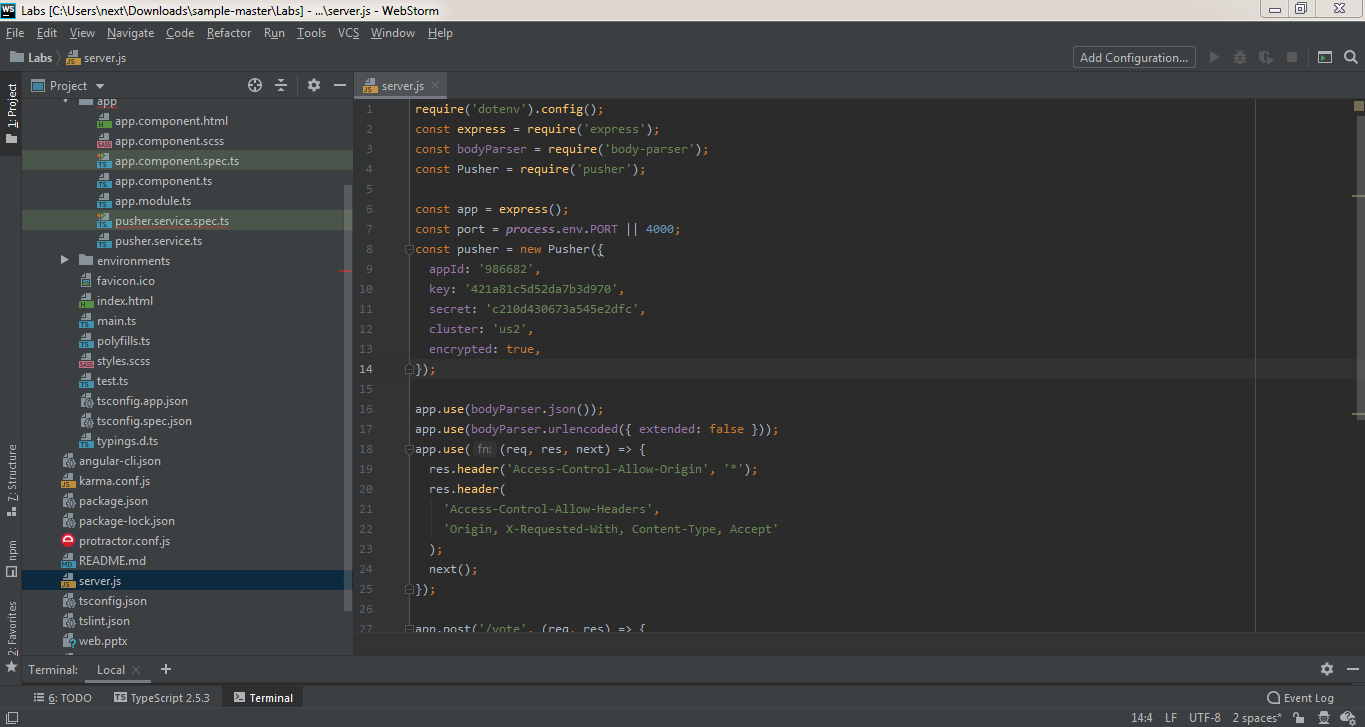


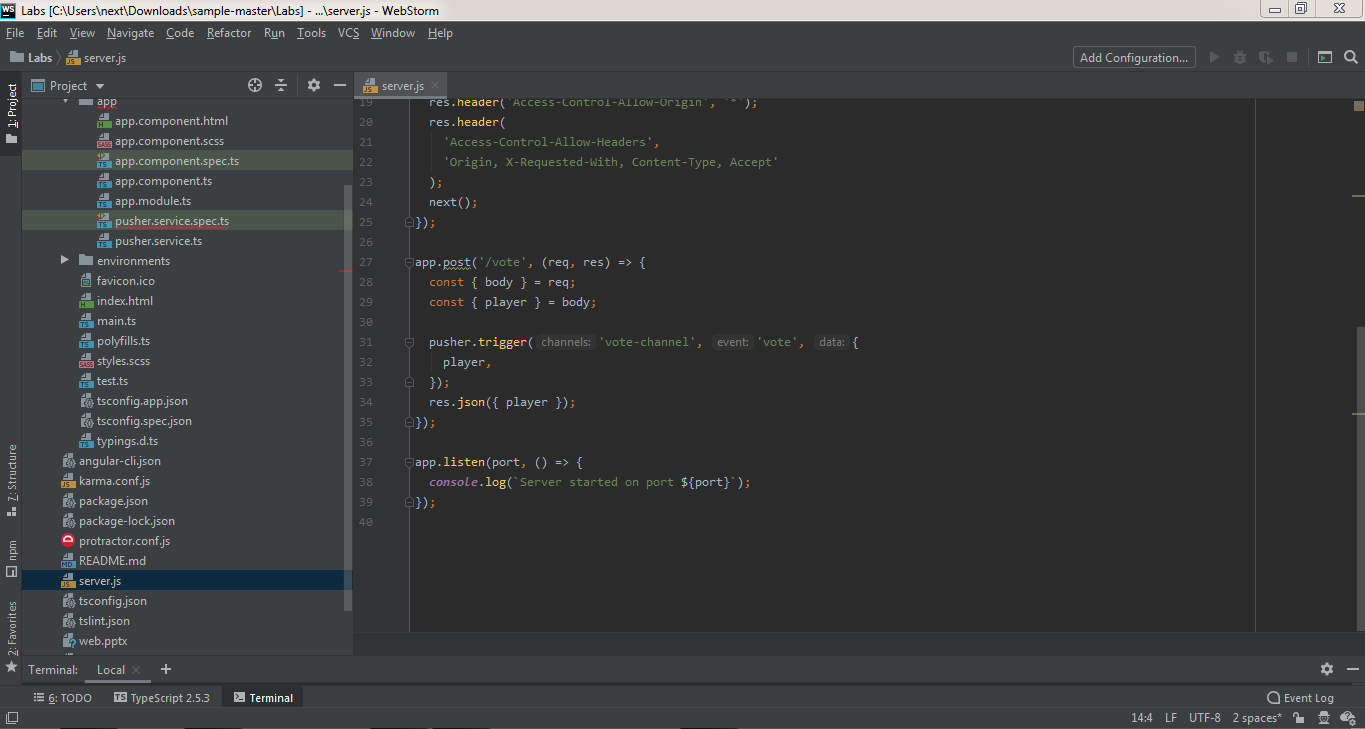
Pusher Service App ID, Key and Cluster Data

index.html file

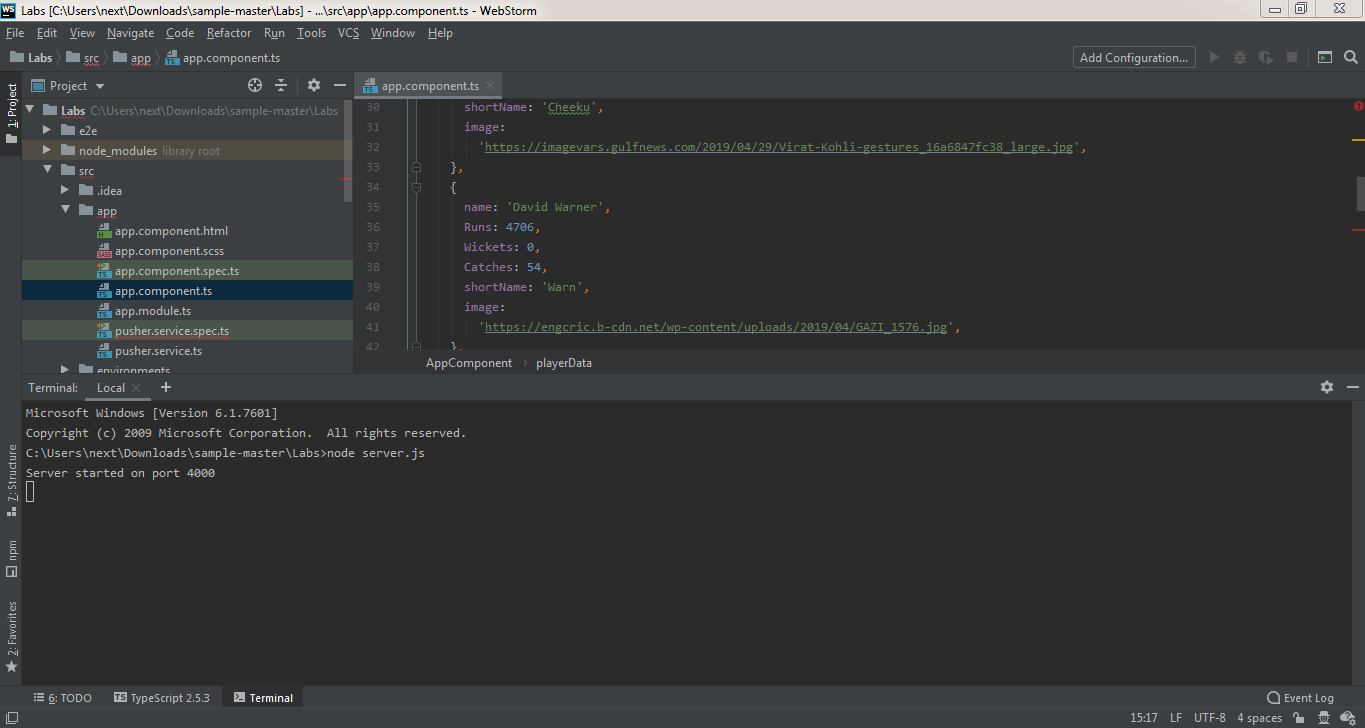


server.js file





**Commands used in the project**



node server.js is used to build the server.

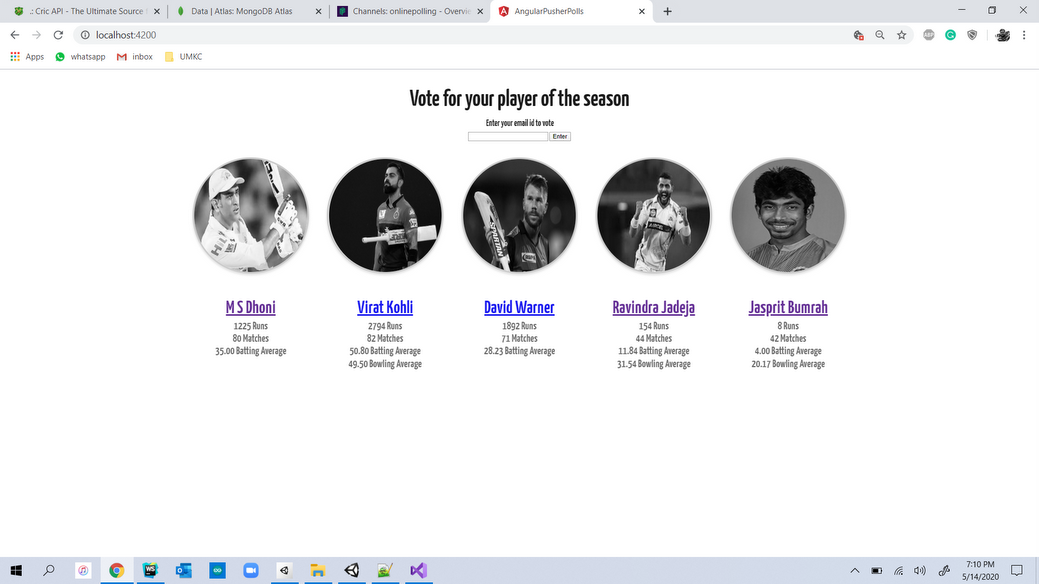


ng serve to build the angular application and serve it locally.

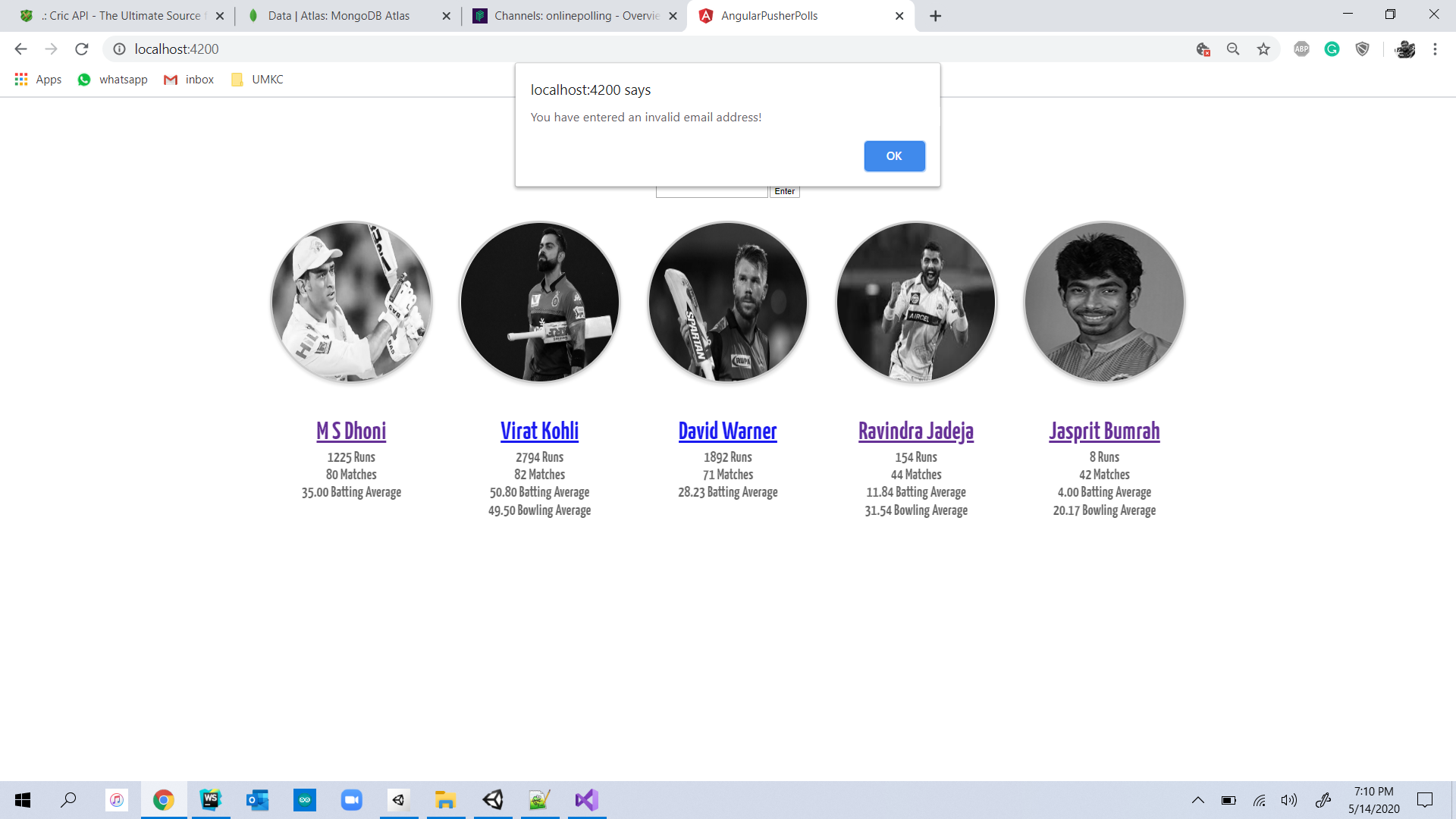
**Working Screens from Project - Output Screens**



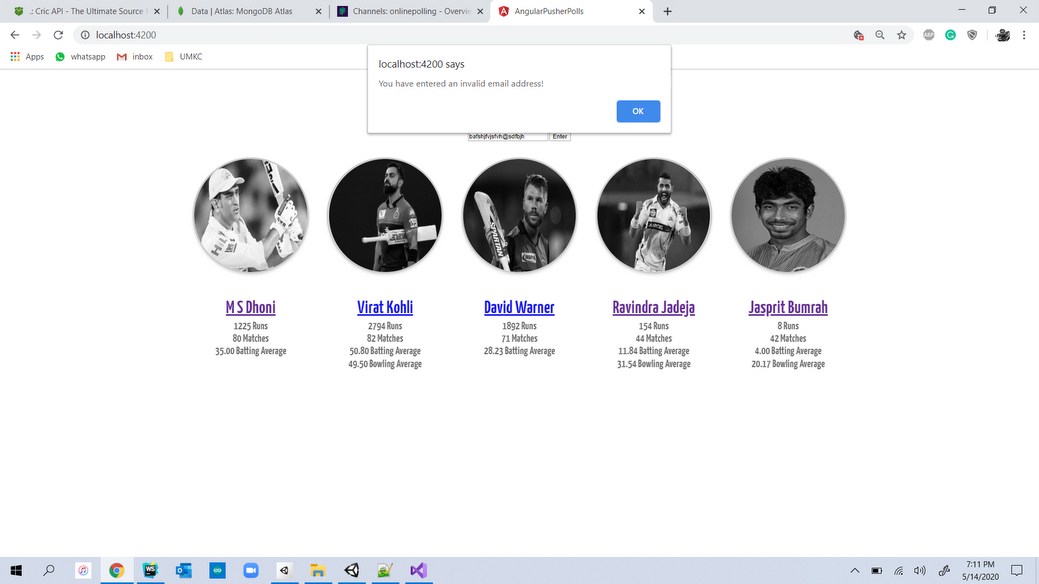
Votes for player in the MongoDB Dashboard.



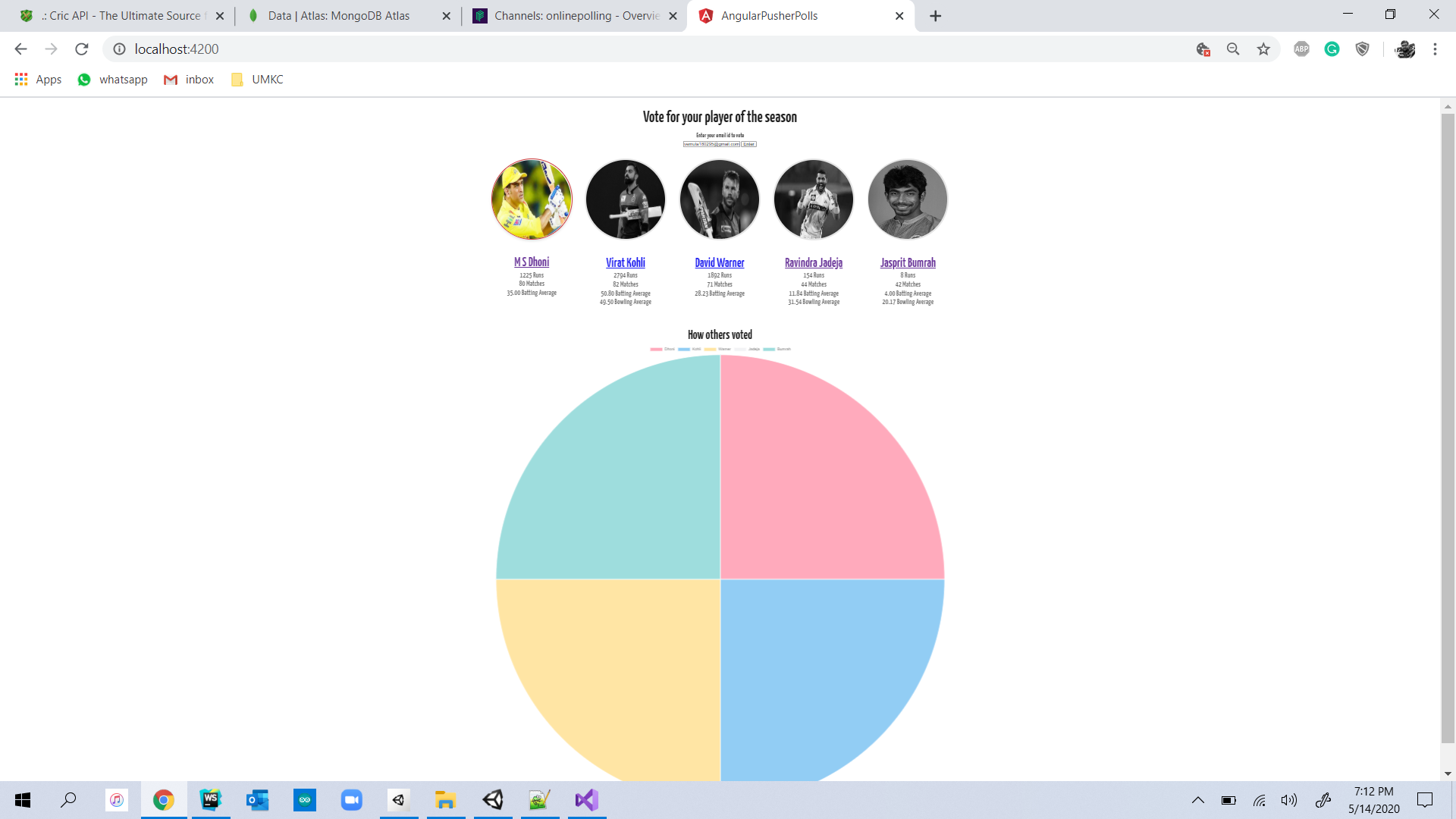
Home Page of Real Time Poll System Application.



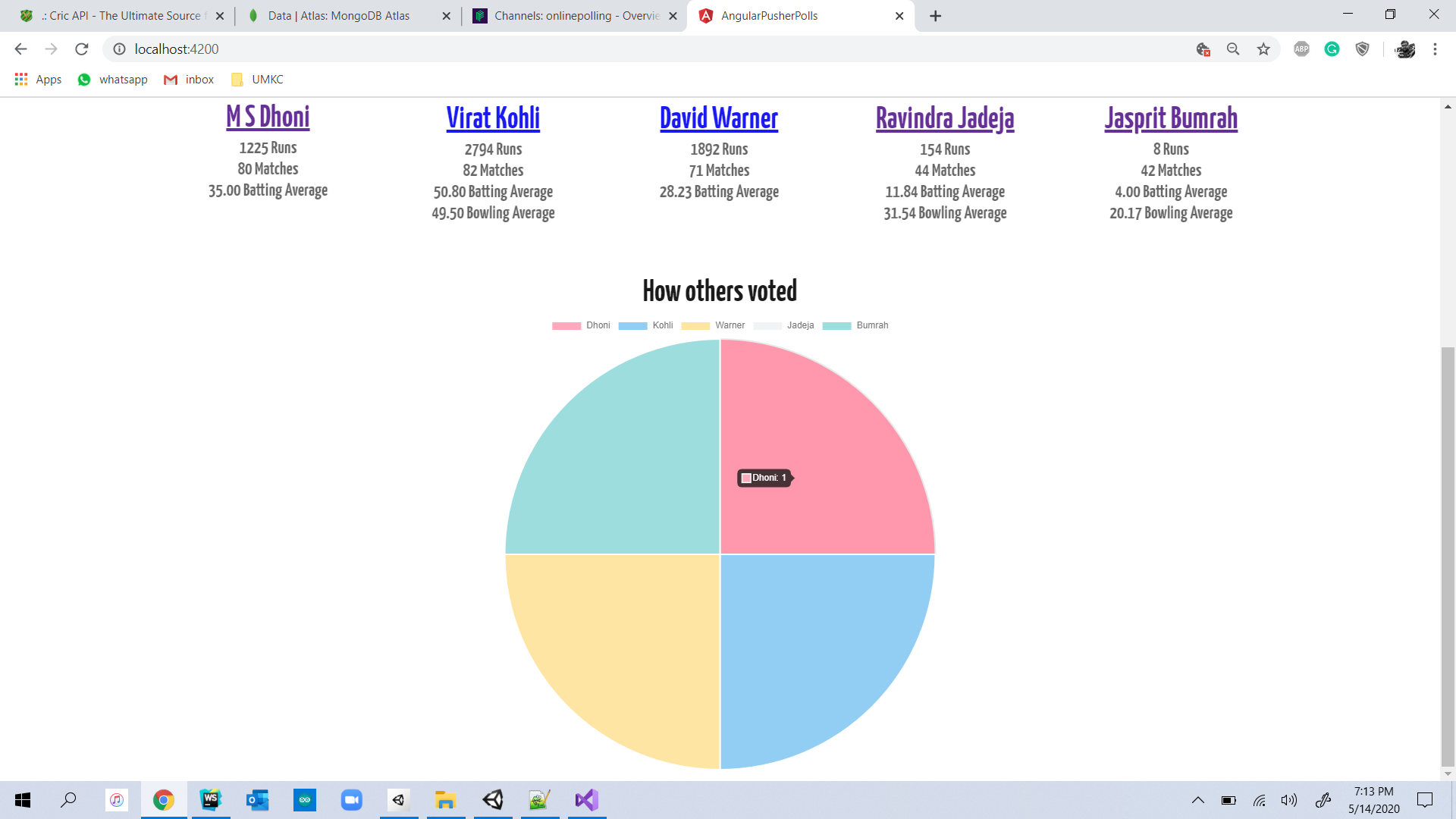
An alert dialog appears when a User tries to vote with blank data in the Email ID column.



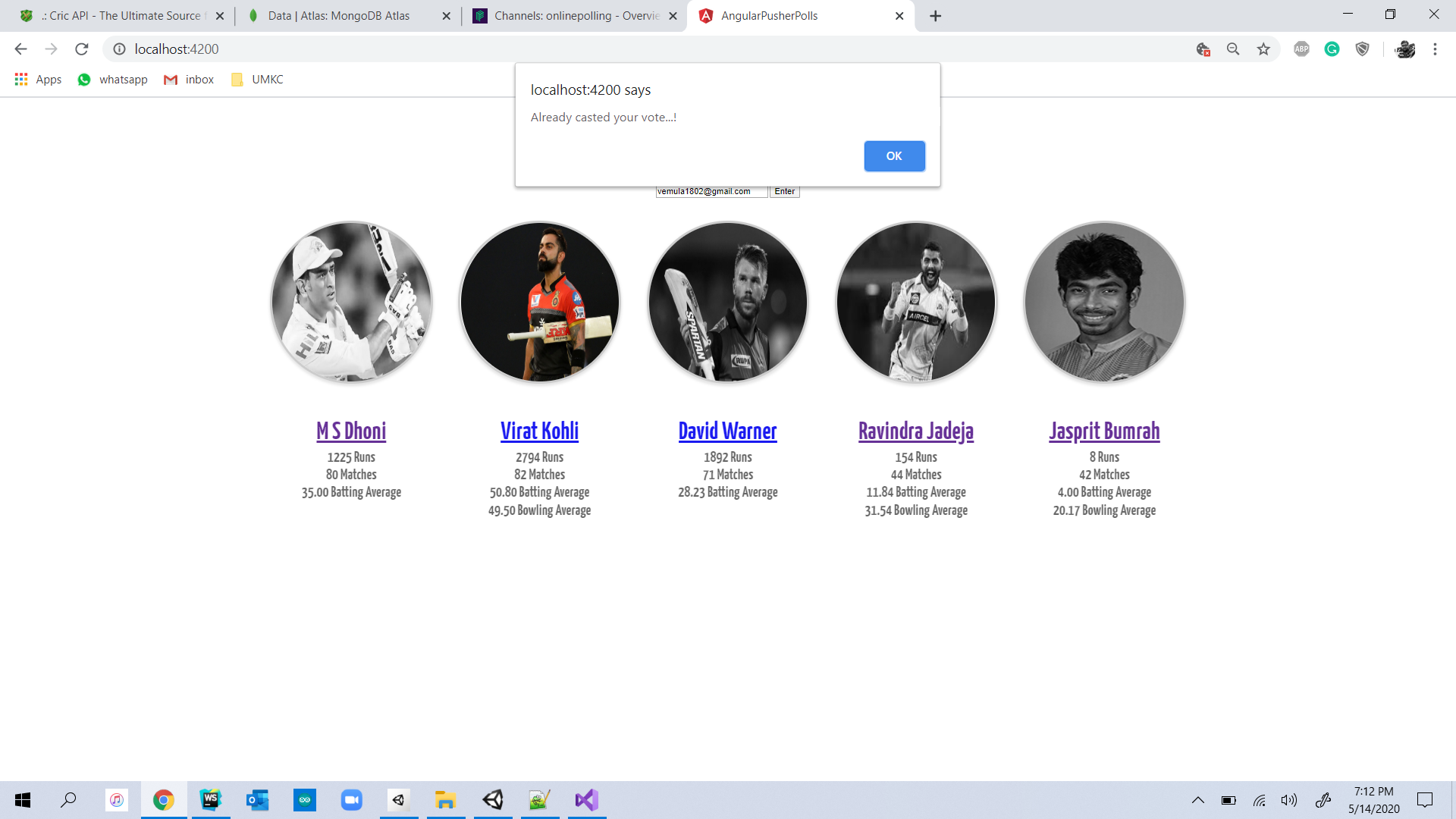
An alert dialog box appears when user tries to cast a vote using an Invalid Email ID.



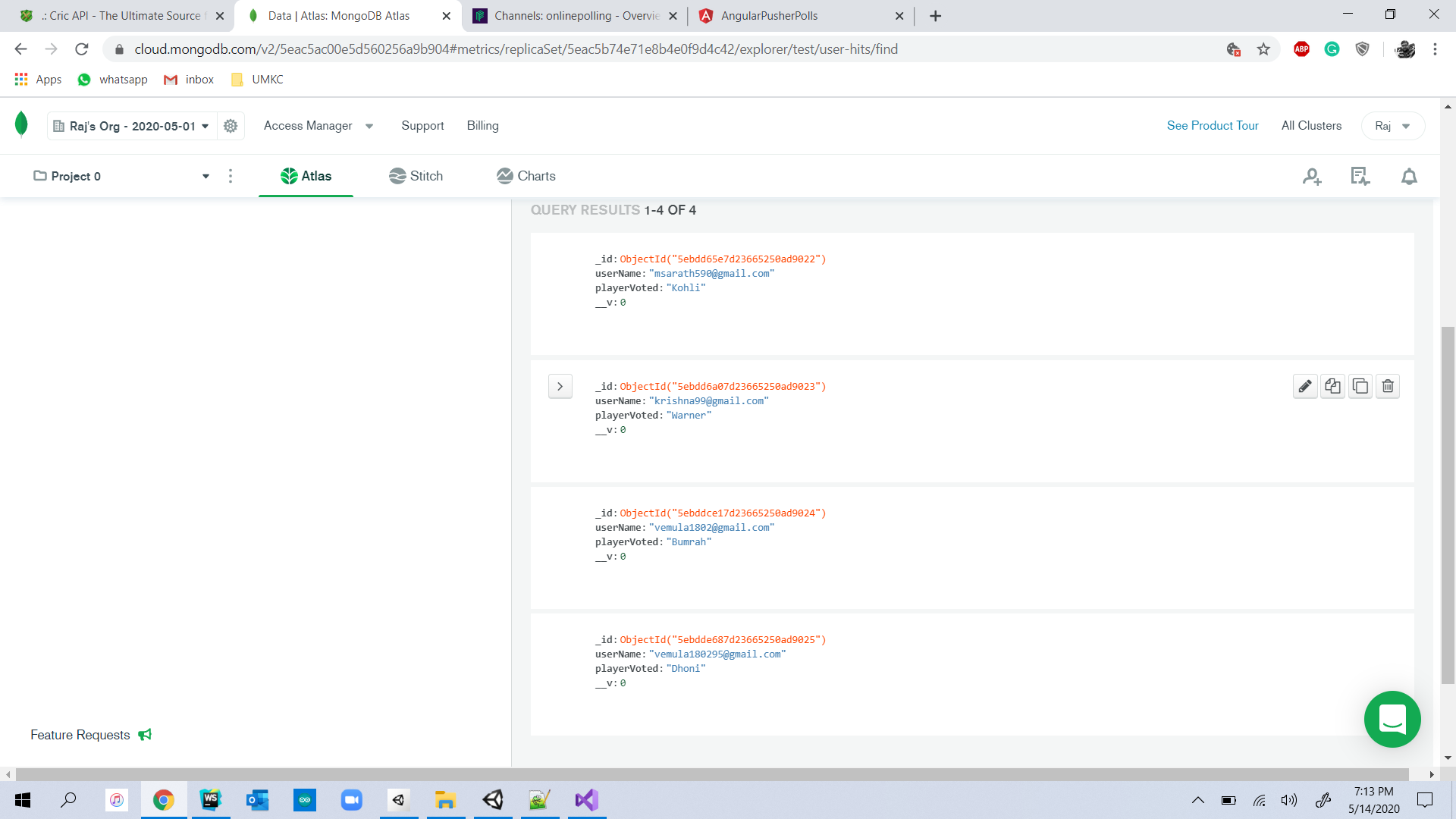
After casting a vote to a player by using a valid Email ID, the voting of each player appears in the form of pie chart. Here the user has casted his vote to player MS Dhoni.



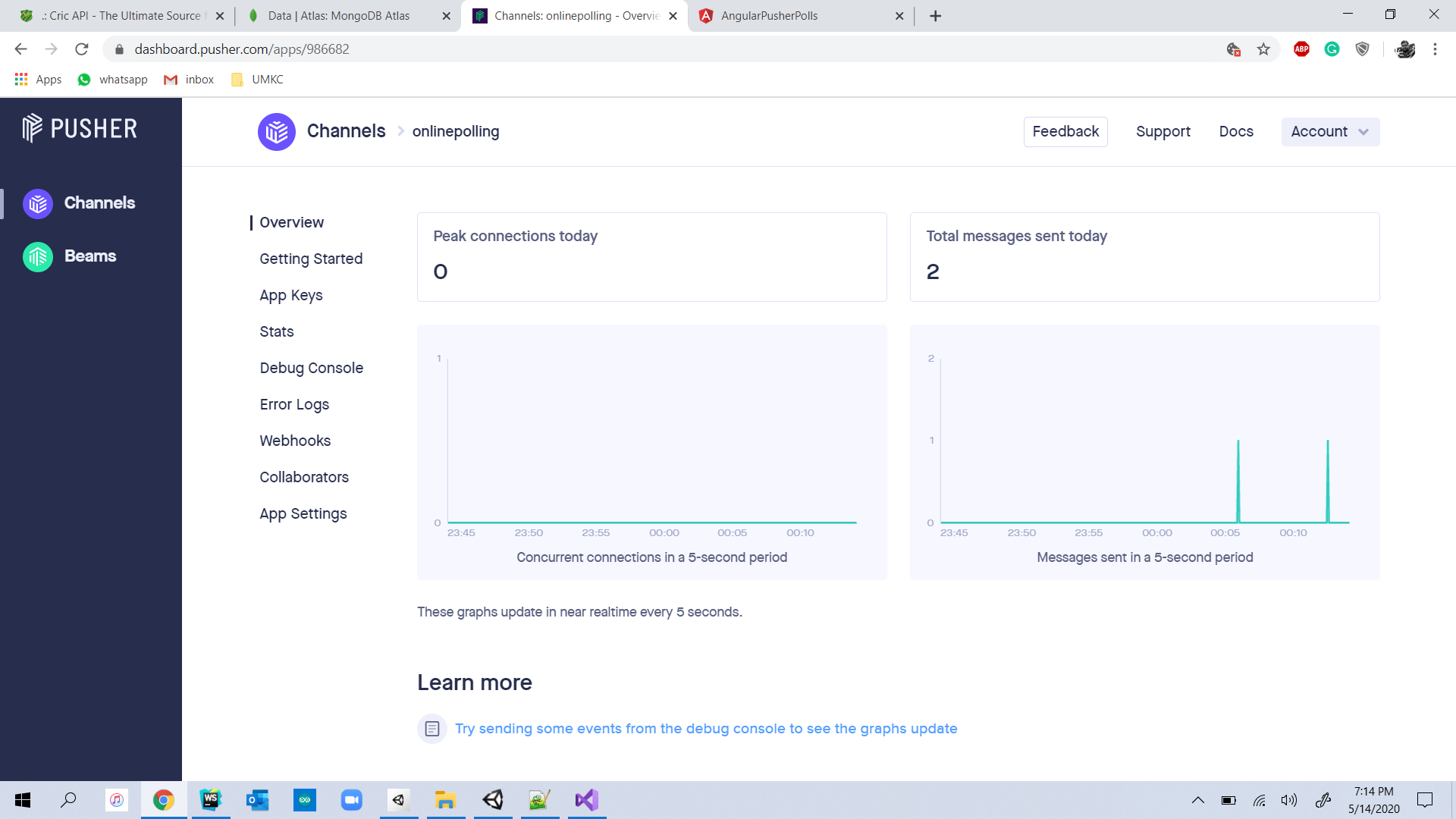
The vote casted to user for MS Dhoni has been displayed in the Pie Chart.



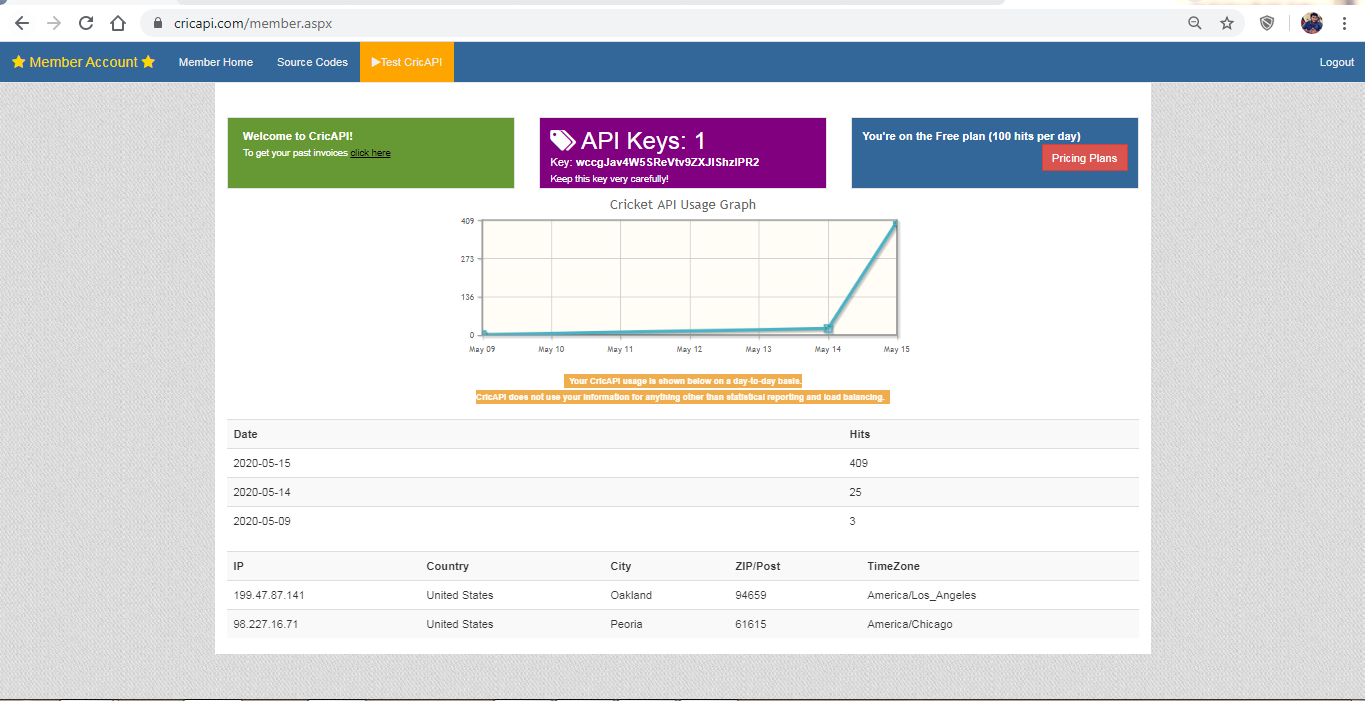
An alert dialog box will appear if user tries to duplicate vote/tries to vote other player using the same Email ID.



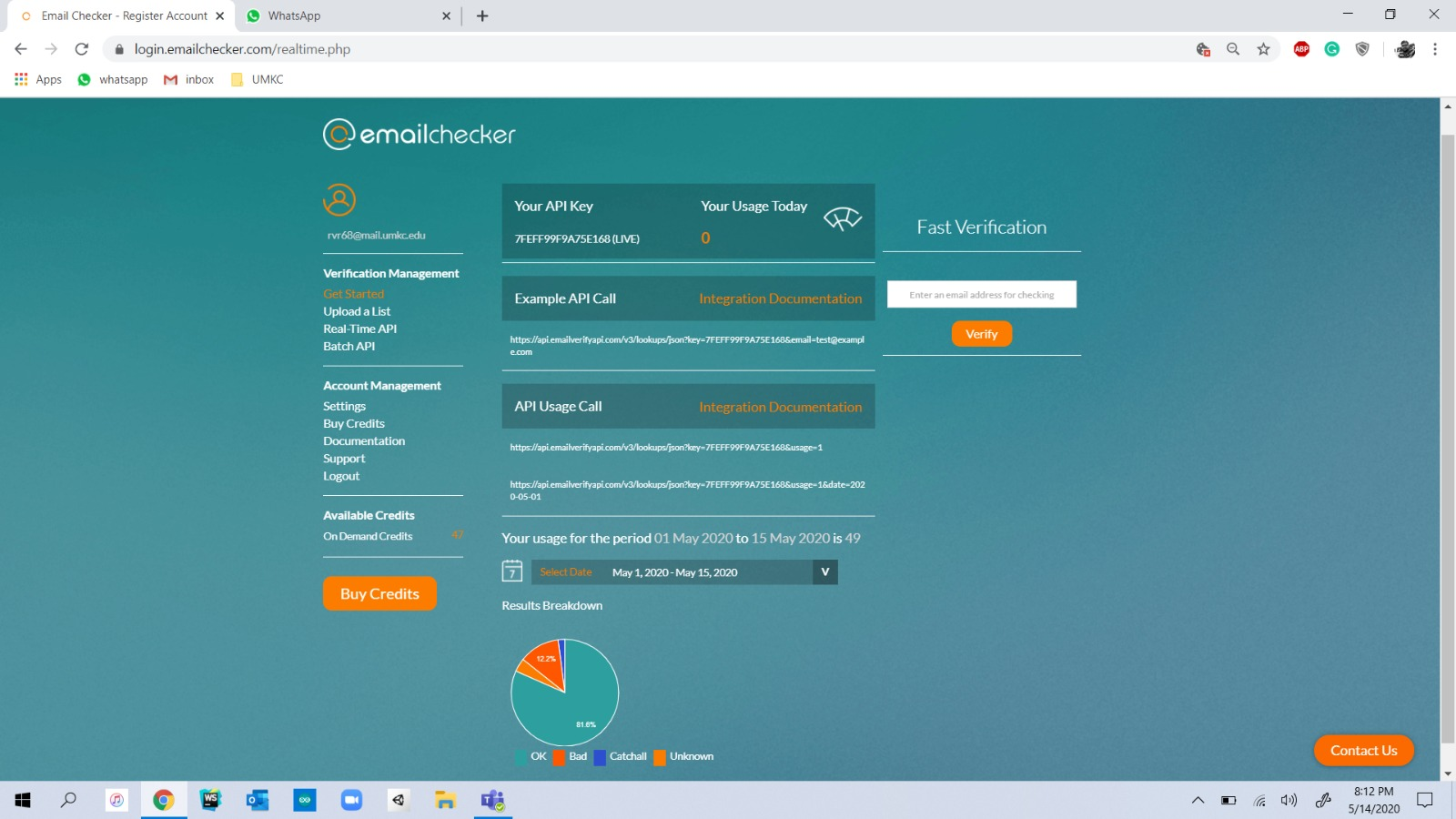
A voting object for player MS Dhoni have been added into MongoDB after successful casting of vote to MS Dhoni by User.



Pusher Dashboard for Real Time Poll System Project



CRIC API Dashboard



Email Checker API Dashboard

**Feedback from Increment – 2**

1. Received a Suggestion to add an Email ID to the application for the verification of user.
2. Update the player statistics using an API service instead of hardcoded values in the code.
3. Remove the duplication of votes by a user.

**Improvement from the Increment - 1**

In the Increment 1 we have just build the application for displaying the data of the player where as in the increment 2 we have developed the code to display an pie chart of vote count of players on voting the favorite player of user and also used pusher service to add the data to the server.

**Improvement from the Increment – 2**

Used API provided by emailchecker.com to validate the Email ID provided by the user. API provided by cricapi.com was used to update the player statistics. Duplication of Votes by a user using a single Email ID has been eliminated by making necessary changes in the code.

**GitHub Link for the Project**

<https://github.com/saikrishnareddykatta/RealTimePollSystem>

**Work/Module Sharing among Teammates**

The work is being shared equally between all the teammates and each one has been provided with a specific task with a timeline.

**Issues and Blockages in the Project**

1. Web Socket Error while accessing the pusher service.
2. Single User has chance to duplicate the votes to their favorite player.
3. Invalid Key Error in pusher service.
4. Difficulties while creating clusters in the MongoDB.
5. Receiving the vote count of each player from MongoDB and representing in the form of a pie chart.

**Future Advancements**

Using a social networking login to login into the application and also to add many other players in different categories. Trying to host the website during an Indian Premier League Season in Year 2021.

**References**

<https://pusher.com/tutorials/>

<https://www.npmjs.com/package/pusher>

<https://www.freecodecamp.org/news/how-to-create-an-online-poll-with-asp-net-core-angular-5-and-highcharts-85ff7fecbaf1/>

<https://www.twilio.com/blog/transfer-files-data-javascript-applications-angular-node-js>

<https://blog.couchbase.com/creating-front-end-user-profile-store-angular-typescript/>

<https://www.cricapi.com/>

https://www.emailchecker.com/