

Introduction

I will be developing a Web Application with CRUD functionalities, using the Vue.js and Node.js frameworks. I plan to do the Web App about the NBA champions last year, the Golden State Warriors. Navbar consisting of home page, players page, and authentication. I will be deploying website in Firebase.

Plan:

- Home page displaying images of GSW players with an add button at the bottom.
- Players page displaying several players, CRUD will be applied in this page i.e.:
 - o Create new Player
 - o Delete Player
 - o Edit Player
- Page for world champions throughout the NBA years, with year, location and team name.
- Authentication (Login, Logout and Signup) using firebase API - Firestore for storing data in the cloud

Set Up:

1. Installed and created Vue.js app in Webstorm
2. Created a Firebase Account
3. Created a Heroku account
4. Created a GitHub account
5. ‘\$ npm install’ to install all necessary dependencies
6. ‘\$ vue add vuetify’ to use Vuertify
7. ‘npm install firebase firestore’

Firebase

I created a new firebase project called “mini-gsw”, I then populated my database using two collections.

Collection	Document ID
players	6L57yUSzyb5YY6zR4TaG
players	eEkTl50I9hjA46fviv4y
players	pHXcbB4ellXgQHhIzyUm
	tRKbUWr1V2x0aByyUxLW

In Firebase I clicked project overview and selected the code button and the following should appear.

```

<script src="https://www.gstatic.com/firebasejs/5.7.2.firebaseio.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyD3tV72d4iKQdjP0dFDXeIsyKl5EmWxwF4",
    authDomain: "mini-gsw.firebaseio.com",
    databaseURL: "https://mini-gsw.firebaseio.com",
    projectId: "mini-gsw",
    storageBucket: "mini-gsw.appspot.com",
    messagingSenderId: "981223853683"
  };
  firebase.initializeApp(config);
</script>

```

I used my own API credentials and created a firebase.js file

```

import firebase from 'firebase'
import 'firebase/firestore'

var config = {
  apiKey: "AIzaSyD3tV72d4iKQdjP0dFDXeIsyKl5EmWxwF4",
  authDomain: "mini-gsw.firebaseio.com",
  databaseURL: "https://mini-gsw.firebaseio.com",
  projectId: "mini-gsw",
  storageBucket: "mini-gsw.appspot.com",
  messagingSenderId: "981223853683"
};

firebase.initializeApp(config);

const db = firebase.firestore();
db.settings({timestampsInSnapshots: true});

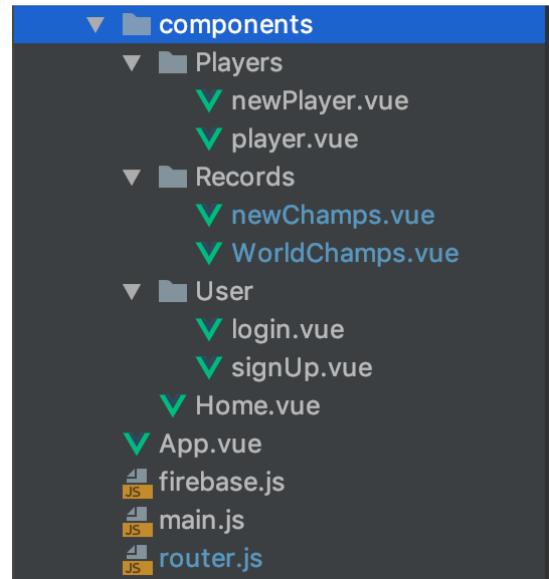
export default db;

```

Webstorm

Created 3 directories and populated those directories as follows:

- Records
 - newChamps.vue
 - WorldChampions.vue
- User
 - Login.vue
 - Signup.vue
- Players
 - newPlayer.vue
 - players.vue

**Every Web Pages****LogIn Page**
SignUp Page

In the Login page user need to enter their registered accounts to be able to view all other Web Pages. All registered users are stored in the Firebase authentication API. Same goes for the signUp page, When the user clicks signUp and fields are valid the data goes straight to firebase.

The screenshot shows the Firebase Authentication interface under the 'Users' tab. It displays a table of three users with the following data:

Identifier	Providers	Created	Signed In	User UID
emmanuel@mail.com		Jan 3, 20...	Jan 3, 20...	LggDeZJrZ3T2lDTATiaTA3...
emmanuel@gmail.com		Jan 4, 20...	Jan 4, 20...	YO2tKS9PTNNXOTfHcJX5...
mawe@gmail.com		Jan 3, 20...	Jan 3, 20...	cJqPU7xieal1CgopCDX6f...

At the bottom, there are pagination controls: 'Rows per page: 50', '1-3 of 3', and navigation arrows.

Home Page once Login.

The screenshot shows the 'GSW HOME' section of the app. At the top, there's a navigation bar with icons for 'HOME', 'VIEW PLAYERS', 'WORLD CHAMPIONS', and 'LOGOUT'. Below the navigation, the title 'Golden State Warriors' is displayed in a large, stylized font. A large image of Klay Thompson in a Golden State Warriors jersey is the central focus. Below the image, there's a placeholder for a new photo with the text 'Add a new home page img' and input fields for 'Image URL' and 'Player Name'. A button labeled 'ADD NEW PHOTO' is at the bottom of the form.

Players Page

Mini-GSW Web App

GSW-HOME

VIEW PLAYERS

WORLD CHAMPIONS

LOGOUT

Golden State Warriors (CRUD)

NBA GSW Players

LeBron James

Height : 6.11 ft
Points Per Game : 30.3
Assist Per Game : 10.1
Steals Per Game : 6.6

Andre Iguodala

Height : 6.11 ft
Points Per Game : 21
Assist Per Game : 0
Steals Per Game : 11

EDIT PLAYER

DELETE PLAYER

EDIT PLAYER

DELETE PLAYER

I have used Vuetify components to make my web app look more appealing. Vuetify helps in structured web pages and more reactive. I used '<v-card>' in the edit functionality so that when we want to edit a player we don't have to open a new web page, it also lessens code and very easy to manage. Delete Alerts:

NBA GSW Players

LeBron James

Height : 6.11 ft
Points Per Game : 30.3
Assist Per Game : 10.1

Player Name
LeBron James

Height
6.11 ft

Points per game
30.3

Assist per game
10.1

Steals per Game
6.6

CANCEL

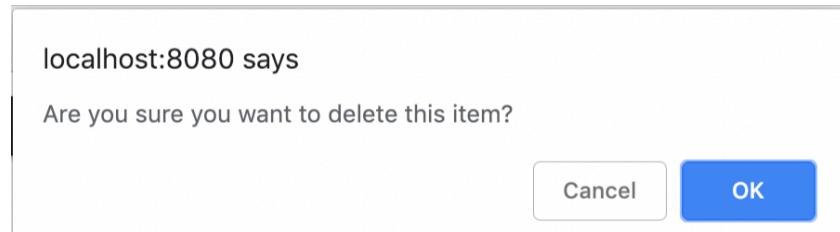
SAVE

Steals Per Game : 11

EDIT PLAYER

DELETE PLAYER

Delete Alerts:



I used alert to inform user before deletion of a player.

When adding new player all fields must be filled, I used the ‘required’ word to do this. I once again used vuetify components in this section. I have two buttons which allows user to save new player or go back to the list of players. It has six text fields:

- - name
- - url
- - height
- - points per game
- - assist per game
- - steals per game

Add a new GSW Player

Player Profile

Legal name* 0 / 30

Img url* 0 / 100

height * ppg * apg * spg *

*indicates required field

BACK

World Records

This pages consist of all the world champions in the NBA. The team name, year of championship and the location of the event. This data is also stored in Firestore. This only consist of two CRUD functionalities, add and delete.

NBA World Championship Records

NBA Champions Records

[ADD NEW RECORD](#)

Team Name	Year of Season ↑	Game Location	Actions
Miami Heat	1996 - 1997	Miami	
Boston Celtics	1997 - 1999	Boston	
Washington Wizards	1999 - 2001	Washington DC	
Houston Rockets	2001 - 2002	Houston Texas	
LA Lakers	2014 - 2015	Los Angeles	

Rows per page:

5

1-5 of 9



Add a new NBA Record

New NBA World Record



Team name*

0 / 30



Year*



Location *

*indicates required field

[BACK](#)[SAVE](#)

Conclusion:

I really enjoyed this project, it was very challenging and it definitely gave so much knowledge about web development. It gave me a overview of how backend and frontend are vital for a responsive web app.

Along the way I had many difficulties, I mainly looked for solutions in stack over flow and video tutorials available online. I have problems with logging in and also with the navbar, I wasn't able to fix when hiding and showing the authentication buttons (Login, SignUp and Logout).

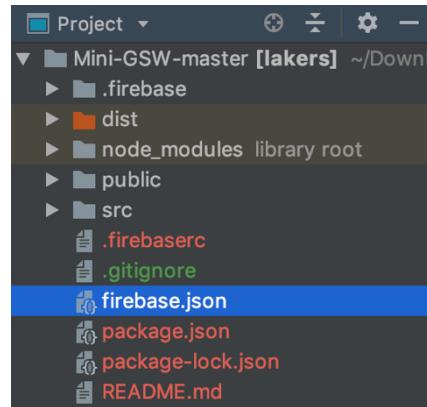
I wanted to hide the signup and login button once login ang only show logout. Also I wanted to hide the logout button at the very start.



Deployment:

1. login into firebase ‘firebase login’.
 2. Create new firebase using ‘firebase init’

3. Run the command ‘npm run build’ and this should create a new folder called ‘dist’.



4. Next enter the command ‘firebase deploy’ to deploy app.

```
ESedicol:Mini-GSW-master esedicol$ firebase deploy --only hosting

== Deploying to 'mini-gsw'...

i  deploying hosting
i  hosting[mini-gsw]: beginning deploy...
i  hosting[mini-gsw]: found 8 files in dist
✓  hosting[mini-gsw]: file upload complete
i  hosting[mini-gsw]: finalizing version...
✓  hosting[mini-gsw]: version finalized
i  hosting[mini-gsw]: releasing new version...
✓  hosting[mini-gsw]: release complete

✓  Deploy complete!
```

I followed the steps for deploying the app into firebase, but I came across problems and weren't able to solve it. When I clicked on the link given I was directed into an empty web page.