



Draumadeild fyrir Amerískan fótbolta

Þorsteinn Sigurðsson



**Iðnaðarverkfræði-, vélaverkfræði- og
tölvunarfræðideild
Háskóli Íslands**

Draumadeild fyrir Amerískan fótbolta

Þorsteinn Sigurðsson

6 eininga ritgerð sem er hluti af
Baccalaureus Scientiarum gráðu í Tölvunarfræði

Leiðbeinandi
Ebba Þóra Hvannberg

Iðnaðarverkfræði, vélaverkfræði- og tölvunarfræðideild
Verkfræði- og náttúruvísindasvið
Háskóli Íslands
Reykjavík, nóvember 2021

Draumadeild fyrir Amerískan fótbolta
Vefsíðu hönnun með React fyrir draumadeildar fótbolta
6 eininga ritgerð sem er hluti af *Baccalaureus Scientiarum* gráðu í Tölvunarfræði

Höfundarréttur © Þorsteinn Sigurðsson
Öll réttindi áskilin

Verkfræði- og náttúruvísindasvið
Háskóli Íslands
Bjargargata 1
102 Reykjavík
Sími: 525 4000

Skráningarupplýsingar:
Þorsteinn Sigurðsson, 2021, *Draumadeild fyrir Amerískan fótbolta*, BS ritgerð,
Tölvunarfræðideild, Háskóli Íslands, 25 bls.

Reykjavík, 17.12 2021

Útdráttur

Amerískur fótbolti fer sívaxandi á Íslandi sem er mestmegnis draumadeildar fótbolta að þakka. Í grunninn er draumadeildarfótbolti stigagjafa leikur og markmiðið í leiknum er að skora fleiri stig heldur en andstæðingurinn þinn. Stigin eru svo metin eftir hvernig leikmenn í amerískum fótbolta í NFL (*e.National Football League*) standa sig í hverri viku. Eitt af vandamálum við draumadeildar fótbolta er að það er erfitt að taka sín fyrstu skref sem byrjandi og gera greinarmun á milli virði hvers og eins leikmanns. Markmiðið mitt með þessu verkefni er að búa til vefsíðu sem væri grundvöllur(*e.platform*) fyrir draumadeildar fótbolta notendur og einnig fyrir þá sem vilja taka sín fyrstu skref í þessum skemmtilega leik. Vefsíðan var hönnuð og forrituð með forritunarsafninu React sem tilheyrir tungumálinu JavaScript. Einnig nýtti ég mér skýjaþjónustur sem Amazon Cloud Service býður uppá fyrir geymslu á gögnum og almennri bakendavinnslu.

Efnisyfirlit

Inngangur	2
1.1 Hvatinn að verkefninu	2
1.2 Meira um draumadeild	2
1.3 Markmiðið með verkefninu	3
1.4 Námsmarkmið	3
2 Þarfagreining	5
2.1 Notendahópar	5
2.2 Notendasögur	6
3 Tækni	7
3.1 JavaScript React	7
3.2 AWS AMPLIFY	8
3.3 BaaS	8
3.4 DynamoDB Gagnagrunnur	9
3.5 GraphQL	9
4 Hönnun og arkitektúr forritalausnir	10
4.1 Hönnun	10
4.2 Gagnaáætlun	10
5 Gagnagrunnur og vefþjónusta	11
5.1 AWS og DynamoDB gagnagrunnur	11
5.2 Sleeper vefþjónusta	11
5.3 GraphQL vefþjónusta	12
6 Forritun	13
6.1 Hooks aðferðafræði	13
6.2 Teikna á DOM tré	13
6.3 Conditional Rendering	14
7 Notkun forrits	15
8 Samantekt	18
8.1 Umræðan um niðurstöður	18
8.2 Áframhaldandi verkefni	18
9 Heimildir	19

Inngangur

1.1 Hvatinn að verkefninu

Amerískur fótbolti er ein stærsta íþrótt í heiminum og fer sívaxandi á Íslandi sem er mestmegnis draumadeildar fótbolta að þakka. Draumdeildar fótbolti er leikur þar sem spilarar geta stofnað deild þar sem hver og einn keppandi velur sér leikmenn í byrjun hvers tímabils og stofnar lið til þess að keppa við aðra notendur í deildinni. Draumadeildin fylgir svo dagskránni í amerískum fótbolta og í hverri viku mætast spilarar í deildinni hvor öðrum og það lið sem endar með flestu stigum í vikunni vinnur leikinn. Í lokin á tímabilinu mætast liðin með flestu sigrana í úrslitakeppni þar sem sigurvegari verður svo loks krýndur.

Eitt það mikilvægast í draumadeildar fótbolta eru leikmannaskipti milli liða í drauma deild. Þetta gerir oft gæfumuninn milli liða sem lenda í fyrsta sæti eða neðarlega í deildinni. Þess vegna vildi ég gera verkefni sem snerist um þessa hlið að draumadeildar fótbolta vegna þess að það er hægt að fara í mikla greiningu á leikmannaskiptum og hvaða áhrif það hefur á lið spilarans.

1.2 Meira um draumadeild

Í byrjun hvers tímabils í draumadeild velur lið sér leikmenn til að stilla upp liði fyrir komandi tímabil. Þetta er gert með svo kölluðu leikmannavali, þar sem hver keppandi er með ákveðna valröð til þess að velja besta leikmanninn sem er í boði hverju sinni. Til eru mismunandi leiðir til að velja leikmenn fyrir draumadeildar liðið. Ein þeirra kallast „Snáka röðun“ þar sem keppandi sem er með fyrsta valrétt í fyrstu umferð velur svo í næstu umferð leikmann í síðasta valrétt. Þessi tegund af draumadeild gerir það að verkum að hver og einn keppandi er með enga sömu leikmenn fyrir komandi tímabil og gerir leikmanna valið jafnt. Deildirnar innihalda oftast 12 keppendur fyrir hvert tímabil. Þetta er algengasta talan en hún verður að vera í sléttri tölu þar sem hver og einn keppandi þarf að mæta öðrum keppanda í deildinni í hverri viku.

Spilarar þurfa að velja kringum 14 leikmenn fyrir komandi tímabil sem samanstendur af u.þ.b. 8 í byrjunarliði og 6 varamenn. Byrjunarliðið fyrir lið í draumadeild samanstendur þá af mismunandi stöðum í Amerísku fótbolta liði en þetta getur verið mismunandi fyrir hverja draumadeild.

Stigagjöf fyrir hvern leikmann í hverri viku fer svo eftir hvernig honum gengur raunverulega í Amerískum fótbolta. Það fer t.d. eftir hversu marga kastaða jarda (*e. Passing yards*), hlaupa jarda (*e. Running yards*), gripna jarda (*e. Recieving yards*) og snertimörk (*e. Touch Down*) sem leikmaðurinn fær. Mikilvægustu leikmennirnir í draumadeildum eru þá stjörnuleikmennirnir í amerískum fótbolta þar sem þeir fá flestu jardana og snerti mörkin.

Fyrir hverja leikviku er svo borin saman stig milli keppanda sem eru að mætast og sá keppandi sem fær flest stig sigrar.

1.3 Markmiðið með verkefninu

Markmiðið mitt með þessu verkefni er að búa til grundvöll (*e.Platform*) fyrir spilara í draumadeildar fótbolta til þess að greina leikmannaskipti og fá yfirlit yfir deildir og lið. Einnig verður til síða til þess að hjálpa byrjendum við að greina muninn á milli virðileikmanna.

Verkefnið í heild sinni verður vefsíða sem mun innihalda upplýsingar um leikmenn, yfirlit yfir deildir og lið notandans og greiningarreiknivél fyrir leikmannaskipti milli tveggja spilara í draumadeild.

1.4 Námsmarkmið

Markmiðið mitt með verkefninu var að læra á forritunarsafnið React sem tilheyrir tungumálinu JavaScript. React er eitt af vinsælustu forritunarsöfnum fyrir hönnun og forritun á framenda. Fyrir bakenda mun ég nýta mér vefþjónustur sem AWS (*e.Amazon Web Services*) býður upp á og gagnagrunns tungumálið GraphQL til þess að hjálpa mér við að vinna með gögnin mín.

Bakgrunnurinn minn fyrir þetta verkefni er að ég hef búið til ýmis forrit sem hafa að hluta til verið forrituð í JavaScript tungumálinu í gegnum áfangana Tölvuleikjaforritun, Tölvugrafik og Vefforritun. Einnig hef ég tekið áfangana Hugbúnaðarverkefni 1, Hugbúnaðarverkefni 2 og Viðmótsforritun sem munu hjálpa mér með vinnslu á framenda og bakenda.

2 Þarfagreining

2.1 Notendahópar

Notendahópurinn fyrir þetta verkefni eru spilarar fyrir draumadeildar fótbolta og fyrir þá sem vilja byrja. Samkvæmt greinum frá FSGA (Fantasy Sports & Gaming Association) voru kringum 50 milljónir draumadeildar spilarar (sjá mynd 2.1) en í þeim hópi eru rúmlega 80% sem spila amerískan draumadeildar fótbolta (Association, 2019). Hér er rétt að benda á að þessi grein fjallaði einungis um þá notendur sem eru í Norður-Ameríku.

FIGURE 2. US-only, 18+ year old fantasy sports players	
Year	Estimated Number of Players
2015	42 Million
2016	42.8 Million
2017	43.2 Million
2019	45.9 Million

Mynd 2.1 Tölur yfir fjölda draumadeildar spilara í öllum íþróttum milli árana 2015-2019 í Norður-Ameríku.

2.2 Notendasögur

Þar sem ég er sjálfur spilarí í draumadeildar fótbolta fannst mér henta vel að búa til nokkrar notendasögur fyrir þarfagreiningu á verkefninu. Ég skilgreindi alls 7 notendasögur í byrjun áfangans og markmiðið mitt var að klára flestar þessar sögur. Alls náði ég að klára allar sögurnar nema notendasögu 3 og 7. Ég er kominn langt með notendasögu 3 en náði ekki að klára hana fyrir lok áfangans.

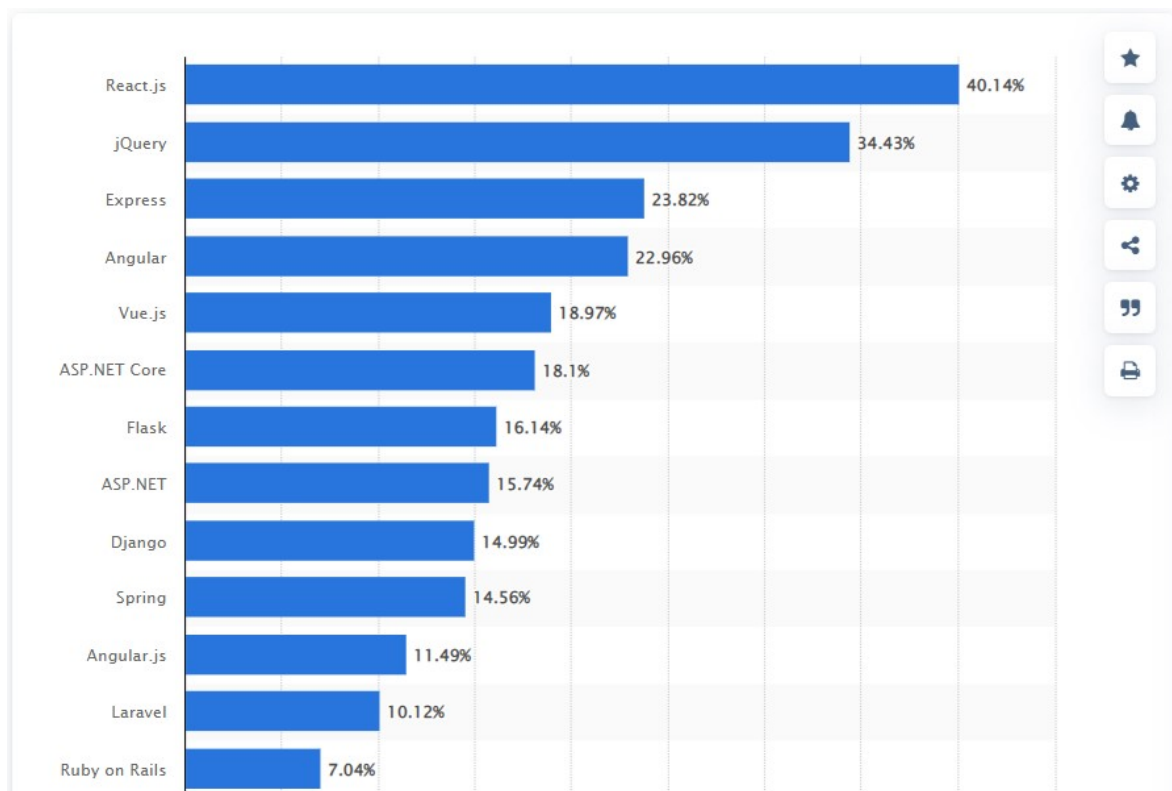
1. As a user I want to login so that I can manage my account.
2. As a user i want to navigate through the website without any difficulties
3. As a user i want to import my team so that i can have a better overview of it.
4. As a user i want to use a trade calculator so that I can determine if a trade is fair.
5. As a user i want to see evaluation from the calculator so that I can get more inside for the trade.
6. As a user i want to see recommendation from the trade calculator so that i can offer a fair trade.
7. As a user i want to get recommendations for which league member to trade with so that I can see what position he needs.

3 Tækni

3.1 JavaScript React

React er framenda JavaScript forritunarsafn sem er notað til þess að byggja gagnvirk viðmót. React forritun snýst aðallega um að nota hluti (*e.components*) sem eru þá teiknaðir í DOM (*e.Document Object Model*) tré hlutann. React býr til hlut af DOM sem kallast virtual DOM. Þessi virtual DOM hlutur er gagnagrind sem inniheldur minni af DOM sem reiknar þá muninn á milli þessa hluta og uppfærir svo DOM eftir því hvernig virtual DOM hefur breyst. Þetta gerir viðmótinu kleift að uppfæra sig á keyrslutíma og þannig fær þetta forritunarsafn nafnið sitt „React“ þar sem þetta er gagnvirk og uppfærir sig á rauntíma. (React, 2021)

Ég ákvað að nota React fyrir þetta verkefni vegna þess að React er eitt af mest notuðu framenda forritunarsöfnum í dag (sjá mynd 3.1) og þar sem ég er með bakgrunn í forritunar-tungumálinu JavaScript fannst mér þetta vera gott tækifæri til að læra á React forritunarsafnið.



Mynd 3.1 Tölfræði yfir mest notuðu framenda forritunarsöfn árið 2021 frá *statista.com* (Statista, 2021)

3.2 AWS AMPLIFY

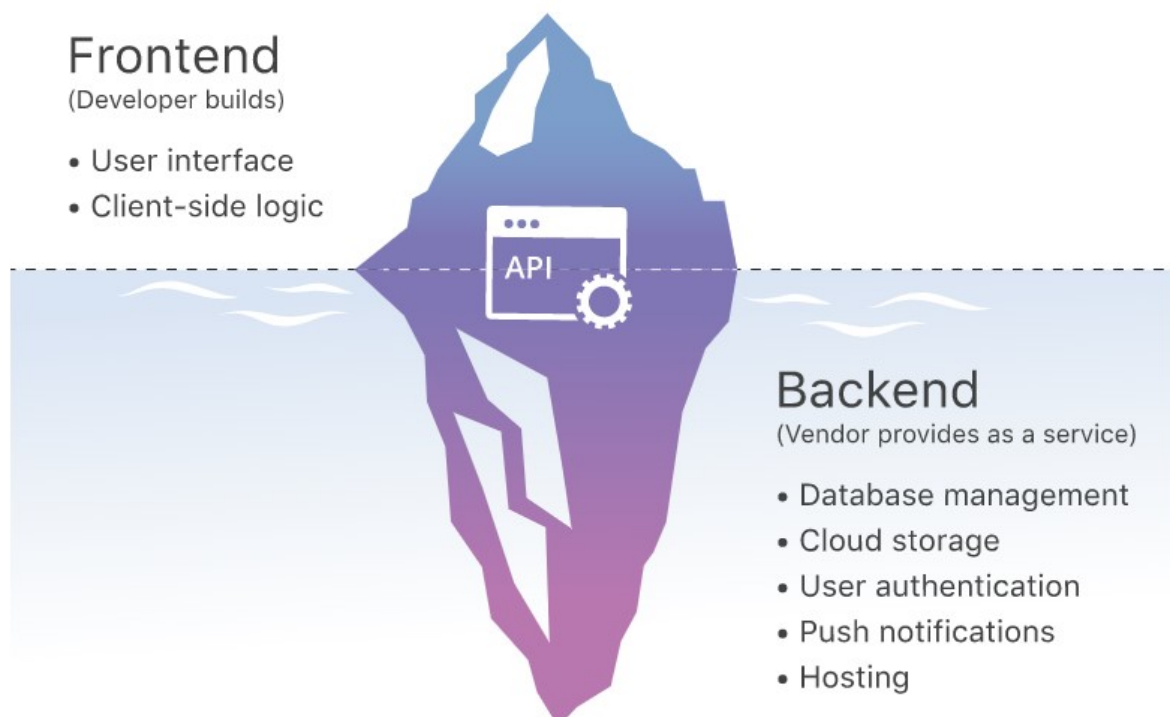
Amplify er AWS (Amazon Web Service) þjónusta á BaaS (Backend as a Service) stigi sem gerir notendum kleift að nýta bakenda þjónustur sem Amazon býður uppá. Bakendinn í Amazon skýinu er í formi DynamoDB töflu þar sem öll gögnin mín fyrir síðuna verða geymd (Amazon, 2021). Með tungumálinu GrapgQL gerir þetta okkur einnig kleift að búa til gagna áætlun fyrir kerfið okkar.

Ég ákvað að velja Amplify fyrir þetta verkefni vegna þess að það býður upp á fullt af gagnarömmum (*e.framework*) eins og notendastýringu, vefhýsingu, tölfræði og gervigreindar pökkum sem ég get nýtt mér fyrir þetta verkefni.

Þar sem ég er í áfanganaum Skýjaforritun og störgögn fannst mér tilvalið tækifæri að kynna mér betur þær skýjaþjónustur sem Amazon býður upp á.

3.3 BaaS

Backend-as-a-Service er skýjaþjónustu módel þar sem notendur þurfa ekki að hugsa um það sem gerist á bakvið tjöldin í bakendanum heldur þurfa þeir einungis að hugsa um framendan. BaaS býður upp á tilbúinn hugbúnað sem sér um alla starfsemi sem á sér stað í net þjónum sem tengjast þá bakendanum (Sjá mynd 3.2).



Mynd 3.2 PaaS módelið sér um alla bakenda vinnuna með tilbúnum hugbúnaði (CloudFlare, 2021)

3.4 DynamoDB Gagnagrunnur

DynamoDB er AWS(Amazon Web Service) gagnagrunnur sem er hannaður til þess að keyra forrit á hvaða skala sem er. Amazon notendur með grunn aðgang fá 25GB af geymslu og kringum 200 milljónir skoðunar eða skriflegar fyrirspurnir á gagnagrunninn í hverjum mánuði. Amazon skýjalausnir bjóða upp á frían aðgang en notendur geta borgað svo aukalega fyrir auka geymslu pláss eða fleiri fyrirspurnir. (AWS, 2021)

DynamoDB gagnagrunnurinn er svo með innbyggt öryggiskerfi þar sem það er reglulega tekið öryggisafrit(*e.Backup*) af gagnagrunninum.

3.5 GraphQL

GraphQL er tungumál til þess að tala við bakenda og gögnin okkar sem eru búin til með Amazon þjónustunni Amplify. Þegar við búum til Amplify forrit er búið til GraphQL þjónustu sem er tengd við bakenda okkar sem býr þá til gagnaáætlun fyrir okkur.

GraphQL er mjög hagkvæmt tungumál fyrir mitt verkefni þar sem ég gat byrjað með mjög einfalda gagnaáætlun í byrjun og gat svo gert hana flóknari með fleiri tengingum þegar lengra leið á verkefnið.

4 Hönnun og arkitektúr forritalausnir

4.1 Hönnun

Hlutirnir sem ég þarf fyrir síðuna væri notandi, leikmenn, lið og deild. Notandi er með one-to-one tengingu í draumadeildar lið og deild sem hann er í. Stefnan er svo að breyta þessu í many-to-one tengingu fyrir áframhaldandi verkefni þar sem notandi getur tengt aðganginn sinn á síðunni fyrir fleiri deildir og lið. Lið notandans er svo með tengingu í leikmanna hlutann með þá one-to-many tengingu þar sem hvert lið samanstendur af mörgum leikmönnum (Sjá mynd 4.2).

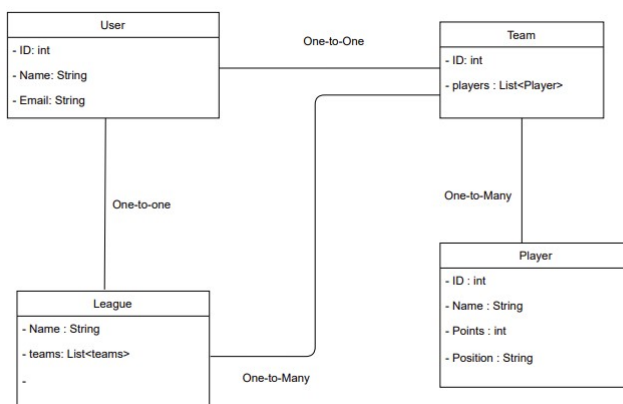
Þetta var grunn áætlunin mín fyrir þetta verkefni þar sem ég myndi byrja einungis með leikmanna hlutann og vinna mig svo áfram þegar sú tenging er komin. Þetta var líka eina sem ég þurfti í byrjun til þess að byrja að forrita skiptingar reiknivélina.

```
amplify-react-app > amplify > backend > api > fantasyAPI > schema.graphql
1  type Player @model{
2    id: ID!
3    name: String!
4    position: String!
5    team: String!
6    points: Int!
7    picture: String
8  }
9
10
```

Mynd 4.1 Dæmi um hvernig leikmann hluturinn okkar er í GraphQL

4.2 Gagnaáætlun

Gagnaáætlunin í formi Class Diagram. Mynd sýnir að user séu með one-to-one tengingu við team og league. League hefur svo one-to-many tengingu við league þar sem deild samanstendur af mörgum liðum. Team hefur svo many-to-one tengingu við Player þar sem Team samanstendur af mörgum leikmönnum.

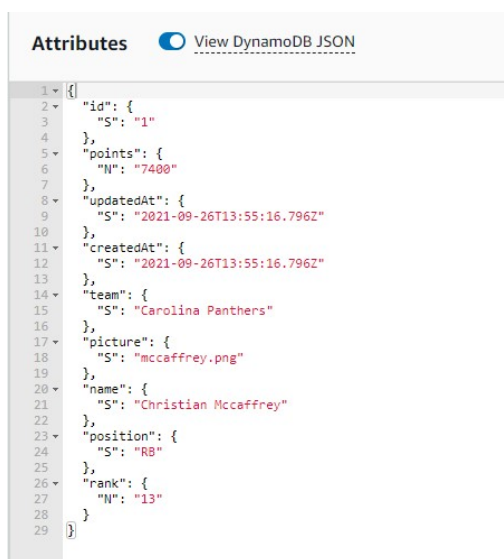


Mynd 4.2 Gagna áætlun (e.data schema) fyrir bakenda.

5 Gagnagrunnur og vefþjónusta

5.1 AWS og DynamoDB gagnagrunnur.

Eftir að ég skilgreindi gagnaáætlunina mína í gegnum GraphQL við uppsetningu á forritinu, var þá búin til DynamoDB tafla fyrir mig sem er tengd við AWS aðganginn minn. Þar get ég annaðhvort búið til gögnin mín handvirkt eða kallað á utanaðkomandi vefþjónustu til þess að sækja gögn. Þar sem ég fann engar góðar netþjónustur til að sækja tilbúin gögn fyrir þá hluti sem ég þurfti, t.d. stig fyrir hvern leikmann, þá bjó ég til mín eigin gögn til þess að geta byrjað að forrita síðuna mína.



Mynd 5.1 Dæmi um röð í töflunni minni á JSON formi fyrir leikmanna hlutann

5.2 Sleeper vefþjónusta

Til þess að sækja upplýsingar um draumadeildar lið og deild spilarans þá nýtti ég mér vefþjónustuna fyrir draumadeildar síðuna Sleeper sem er ein mest notaða draumadeildar fótbolta vefsíðan. Þar get ég kallað á vefþjónustuna til sækja upplýsingar um lið, deildir og margt fleira.

Ég er ekki búinn að forrita þessa tengingu við gagnaáætlunina mína en það verður stefnan með áframhaldandi verkefni.

Dynasty Sandwich	PlayerAPI.js:77
NFL Dynasty Ísland	PlayerAPI.js:78
Flotta NFL deildin	PlayerAPI.js:79
>	

Mynd 5.2 Dæmi um vefþjónustu kall til að sækja deildir fyrir notendanafn á sleeper.com.

5.3 GraphQL vefþjónusta

Til að senda fyrirspurnir á bakenda okkar notum við GraphQL tungumálið. Við uppsetningu á bakenda eru búnar til nokkrar tilbúnar fyrirspurnir, t.d. sækja leikmann eftir ID og sækja alla leikmenn sem við notum þá til að eiga samskipti við gögnin okkar (Sjá mynd 5.3).

```
export const listPlayers = /* GraphQL */ `
  query ListPlayers(
    $filter: ModelPlayerFilterInput
    $limit: Int
    $nextToken: String
  ) {
    listPlayers(filter: $filter, limit: $limit, nextToken: $nextToken) {
      items {
        id
        name
        position
        team
        points
        picture
        createdAt
        updatedAt
      }
      nextToken
    }
  }
`;
```

Mynd 5.3 Dæmi um fyrirspurn til að sækja alla leikmenn.

```
const playerData = await API.graphql(graphqlOperation(listPlayers))
const playerList = playerData.data.listPlayers.items;
```

Mynd 5.4 Geyma alla leikmenn í breytu með því að kalla á GraphQL vefþjónustuna.

Svo er hægt að bæta við fyrirspurnir í vefþjónustunni okkar sem ég þarf fyrir áframhaldandi verkefni fyrir t.d. að sækja leikmenn eftir liði og deildum.

6 Forritun

6.1 Hooks aðferðafræði

Hooks aðferðafræði snýst um að halda utan um ástand á gögnum. Þetta er gert með því að kalla á GraphQL þjónustuna okkar til að sækja ástandið á gögnunum og geyma þau í fylki. Upprunalegu gögnin eru þá geymd í fylki á meðan ástandið á gögnunum er geymt í öðru fylki. Þetta gerir okkur kleift að uppfæra gögnin okkar á keyrslutíma.

```
//Breytur fyrir players gögnin mín
const [players, setPlayers] = useState([]);

useEffect(() =>{
  fetchPlayers();
}, []);

const fetchPlayers = async () => {
  try{
    const playerData = await API.graphql(graphqlOperation(listPlayers))
    const playerList = playerData.data.listPlayers.items;
    setPlayers(playerList);
  }catch(err){
    console.log('error on fetching players', err);
  }
}
```

Mynd 6.1 Dæmi um „hooks“ aðferð til að sækja núverandi ástand (*e.state*) á gögnunum.

6.2 Teikna á DOM tré

Til þess að teikna hluti á DOM tréð skilum við alltaf þeim hlutum við viljum teikna. Þetta virkar þá eins og hefðbundinn HTML stigveldi (*e.hierarchy*) og teiknum þá á virtual DOM tréð okkar sem ber svo saman við DOM tréð og teiknar þá hluti sem við skilum.

```
return(
  <div className = "App">
    { /*Fellilisti til þess að raða eftir dálkum*/ }
    <div className = "header_option">
      <h5>Velja leikmann eftir:</h5>
      <select onChange={ (e) => setSortType(e.target.value)}>
        <option value="name">Nafni</option>
        <option value="position">Stöðu</option>
        <option value="team">liði </option>
        <option value="points">Stigum</option>
      </select>
    </div>
  </div>
)
```

Mynd 6.2. Dæmi um kóða til þess að teikna hluti á DOM tréið. (finna betra dæmi)

6.3 Conditional Rendering

Það sem ég notaði mikið í verkefninu mínu var að teikna á DOM tréð eftir ástandi á breytum. Þetta virkar alveg eins og ástand (*e.Condition*) í JavaScript þar sem við teiknum aðeins þá hluti sem við viljum að teiknist á DOM tréð (sjá mynd 6.3). Ég notaði þessa aðferð sérstaklega fyrir skiptingar reiknivélina þar sem ég vildi teikna allskonar tölfræði hluti sem tengjast þá skiptingunni milli leikmanna. Var með allskonar breytur sem skilgreindu hvort skiptin voru sanngjörn eða ósanngjörn og gat þá teiknað hluti tengda við ástandið á þeim breytum.

```
function renderButtons(){
  if(heildarMismunur != 0){
    return(
      <div className = "ButtonsTrade">
        <Button variant = "contained" onClick = {resetAll}>Hreinsa</Button>
        <Button variant = "contained">Nánari greining</Button>
      </div>
    );
  }
}
```

Mynd 6.3. Dæmi um kóða til þess að teikna á DOM tréið eftir ákveðnu ástandi

7 Notkun forrits

Hér á næstu myndum mun ég sýna virkni af vefsíðunni.

Sign in to your account

Username *





Password *

Forgot your password? [Reset password](#)






No account? [Create account](#)

SIGN IN


Mynd 7.1. Innskráning notenda og stofnun á aðgangi. Notendasaga 1.

Fantasy				
Rankings			Notandi	SIGN OUT
Trade reiknivél			thorsteinnsig@hotmail.com	
API köll				
Velja leikmann eftir: <input type="text" value="Nafni"/>				
<input type="text" value="leita eftir leikmanni"/>				
TJ Hockenson	TE	Detroit Lions	3750	
Tom Brady	QB	Tampa Bay Buccaneers	3700	
Leonard Fournette	RB	Tampa Bay Buccaneers	5300	
Damien Harris	RB	New England Patriots	4600	

Mynd 7.2. Heimasiða með yfirlit á leikmönnum. Nafn, staða, lið, stig og mynd af leikmönnum. Notenda saga 2.

Rankings Trade reiknivél API köll		Fantasy		Notandi thorsteinnsg@hotmail.com	SIGN OUT
Velja leikmann eftir: Stigum		leita eftir leikmanni			
Derrick Henry	RB	Tennessee Titans	10000		
Alvin Kamara	RB	New Orleans Saints	9900		
Dalvin Cook	RB	Minnesota Vikings	9800		
Najee Harris	RB	Pittsburgh Steelers	9700		
Ezekiel Elliot	RB	Dallas Cowboys	8700		

Mynd 7.3. Raða leikmönnum eftir stiga fjölda. Notenda saga 2.

Rankings Trade reiknivél API köll		Fantasy		Notandi thorsteinnsg@hotmail.com	SIGN OUT
Velja leikmann eftir: Stigum		davante			
Davante Adams	WR	Green Bay Packers	8600		

Mynd 7.4. Leita eftir nafn á leikmanni með leitunarvél. Notenda saga 2.

Lið 1

X

 stig:

X

 stig:

X

 stig:

Lið 2

X

 Stig:

X

 Stig:

X

 Stig:

+ Bættu við leikmönnum hjá báðum liðum

Mynd 7.5. Yfirlit yfir Skiptingarreiknivélina þar sem notendur geta valið leikmenn úr lista eða leitað eftir nafni. Notenda saga 4.

Lið 1

X stig: 5300
 X stig:
 X stig:

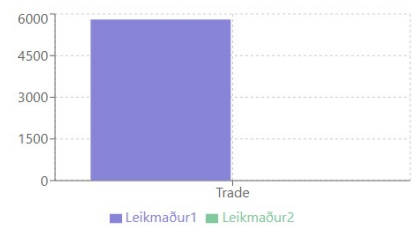
Lið 2

X Stig:
 X Stig:
 X Stig:

< Lið 1 vinnur skiptin með 5800 stigum

Leikmenn til að jafna skiptin

Leikmaður	Stig
Mike Evans	5450
Deebo Samuel	6000
DK Metcalf	5600
DJ Moore	5700
Chris Godwin	5350



HIREINSA

NÁNARI GREINING

Mynd 7.6. Notandi velur leikmann fyrir lið 1 og reiknivélin birtir lista af leikmönnum til að jafna út skiptin. Notenda saga 4,5 og 6.

Lið 1

X stig: 5300
 X stig:
 X stig:

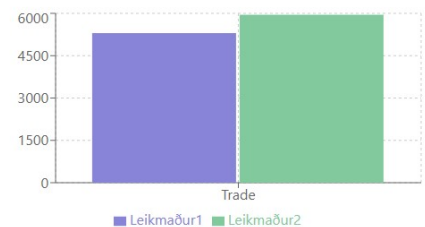
Lið 2

X Stig: 5450
 X Stig:
 X Stig:

> Jöfn skipti

Leikmenn til að jafna skiptin

Leikmaður	Stig
-----------	------



Mynd 7.7. Notandi velur leikmann fyrir lið 2 og jafnar út skiptin. Notendasaga 4,5 og 6

8 Samantekt

8.1 Umræðan um niðurstöður

Markmiðið mitt með þessu verkefni var að gera grundvöll(*e.platform*) fyrir draumadeildar fótbolta spilara í formi skiptingar reiknivélar og leikmanna yfirlit. Ég náði ekki að klára allt sem ég vildi sem tengdist gagnaáætluninni minni, meðal annars lið og deildir notandans. En það helsta sem ég tók út úr þessu verkefni var að þetta fer allt eftir gögnunum sem ég er með, þar sem reiknivélin þarf að hafa einhver stig bakvið hvern og einn leikmann eða einhverskonar kerfi sem reiknar sjálfkrafa stigin fyrir leikmanninn. Það voru engar vefþjónustur(*e.API*) eða gagnagrunnar sem ég fann á netinu sem buðu upp á þetta þannig ég þurfti að búa til öll gögnin sjálfur sem er ekki gott fyrir utanumhald á vefsíðu. Þetta er þá hindrun fyrir áframhaldandi verkefni og þarf þá helst að finna betri vefþjónustur eða gagnagrunn til að sækja þau gögn sem ég þarf á að halda.

8.2 Áframhaldandi verkefni

Framhaldið á þessu verkefni væri að lagfæra stiga kerfið fyrir leikmanna hlutann og klára tengingu milli notendur og lið/deildir sem ég var byrjaður á. Hægt er svo að nýta gervigreindar eða tölfraði pakkana sem Amplify býður upp á til þess að útbúa stigakerfi fyrir leikmenn eða finna betri vefþjónustur sem innihalda gögnin sem ég er að leitast eftir.

Github hlekkurinn fyrir þetta verkefni er að finna hér:

<https://github.com/th251/Lokaverkefni>

9 Heimildir

Amazon. (2021). *AWS Amplify*. Amazon. <https://aws.amazon.com/amplify/>

Association, F. S. (2019). *Industry Demographics*. Middleton: Association, Fantasy Sports & Gaming. <https://thefsga.org/industry-demographics/>

AWS. (2021). *Amazon DynamoDB*. Amazon Web Service.

<https://aws.amazon.com/dynamodb/>

CloudFlare. (2021). *What is BaaS*. CloudFlare.

<https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>

FSGA. (2019). *Industry Demographics*.

Rovere, M. (2018). According to FSGA, Fantasy sports players at an all-time high. *The MediaPlex*, 1. <http://themediaplex.com/according-to-fsga-fantasy-sports-players-at-an-all-time-high/>

Statista. (2021). *Most used web frameworks among developers worldwide, as of 2021*. Statista. <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>

React. (2021). *A JavaScript loprary for building user interfaces*. Reactjs

<https://reactjs.org/>