



## PROJECT SOCCERSCORE

วิชา Advanced Web Technology and Protocols รหัสวิชา 060213742

### จัดทำโดย

นายวุฒิพงศ์	จันทร์น้ำ	รหัสนักศึกษา 61-060216-1202-5
นายกฤษฎีกา	วงศ์สุบรรณ	รหัสนักศึกษา 61-060216-1119-3
นายธนธรณ์	เปรมปราณี	รหัสนักศึกษา 61-060216-1221-1
นางสาวเสาวรจน์	สมเหมาะ	รหัสนักศึกษา 61-060216-1207-6
นางสาวกรพินธุ์	อินทร์เฟื่อง	รหัสนักศึกษา 61-060216-2201-2

### เสนอ

อาจารย์อนิราศ มิ่งขวัญ

ภาคเรียนที่ 2/2563

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา AdvancedWeb Technology and Protocols รหัสวิชา 060213742 โดยมีจุดประสงค์เพื่อรายงานผลการทำโปรเจค SOCCERSCORE ที่เป็นการดึง API จากภายนอก มาใช้งานในการทำโปรเจค มีการใช้ Mongoddb มาเก็บข้อมูลโดยได้มีการนำความรู้ที่ได้เรียนมา มาประยุกต์ใช้ในการทำโปรเจคนี้อีกด้วย

ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่า รายงานฉบับนี้จะมีประโยชน์ไม่มากก็น้อยและถ้าหากว่ามีข้อผิดพลาดประการใดทางคณะผู้จัดทำก็ต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

26/04/2564

## ขั้นตอนการติดตั้งโปรเจกต์ SOCCERSCORE

1. ติดตั้ง Python
2. ติดตั้ง Node.js
3. ติดตั้ง MongoDB ถ้ามี Database อยู่บน Cloud ก็ไม่ต้องติดตั้ง
4. ทำการ Clone Project จาก GitHub เมื่อ Clone เสร็จแล้ว ให้ทำการใช้คำสั่งดังนี้
  - npm -v (เอาไว้เช็คว่ามี npm แล้ว)

```
PS D:\บี3\จอม2\AdvancedWeb\soccerscore> npm -v
6.14.12
```

- npm install -g @vue/cli

```
PS D:\บี3\จอม2\AdvancedWeb\soccerscore> npm install -g @vue/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
C:\Users\11\AppData\Roaming\npm\vue -> C:\Users\11\AppData\Roaming\npm\node_modules\@vue\cli\bin\vue.js
+ @vue/cli@4.5.12
updated 1 package in 44.044s
```

- npm i

```
PS D:\บี3\จอม2\AdvancedWeb\soccerscore> npm i
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
audited 1376 packages in 10.458s
```

- npm run serve

```
PS D:\บี3\จอม2\AdvancedWeb\soccerscore> npm run serve

> soccerscore@0.1.0 serve D:\บี3\จอม2\AdvancedWeb\soccerscore
> vue-cli-service serve

INFO Starting development server...
98% after emitting CopyPlugin

DONE Compiled successfully in 3452ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.1.10:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

## 5.ทำการ Clone api เมื่อ Clone เสร็จแล้ว

- `pip install -r .\requirements.txt`

```
PS D:\311\venv2\AdvancedWeb\Project\soccerAPI> pip install -r .\requirements.txt
Requirement already satisfied: Flask==1.1.2 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfk
te-packages (from -r .\requirements.txt (line 1)) (1.1.2)
Collecting pymongo[srv]
  Downloading pymongo-3.11.3-cp39-cp39-win_amd64.whl (383 kB)
    |#####| 383 kB 2.2 MB/s
Collecting Flask_Cors==3.0.10
  Downloading Flask_Cors-3.0.10-py2.py3-none-any.whl (14 kB)
Collecting pytz==2021.1
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
    |#####| 510 kB 6.8 MB/s
Collecting requests==2.25.1
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
    |#####| 61 kB 3.8 MB/s
```

---

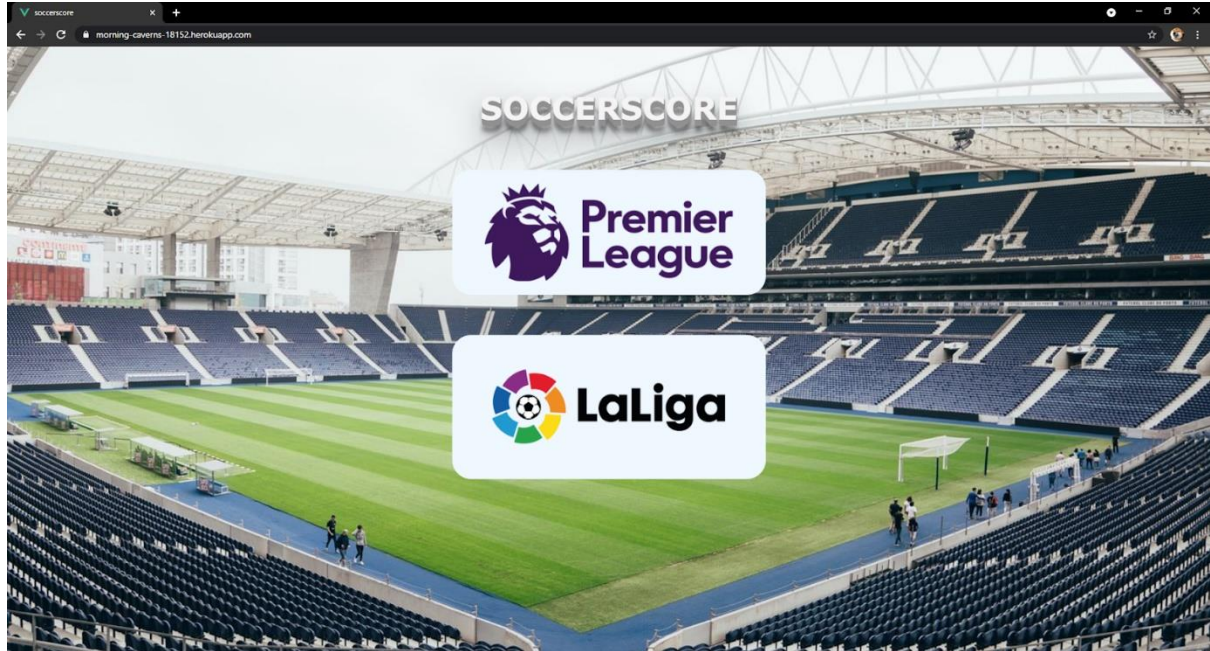
- `python ./app.py`

```
PS D:\311\venv2\AdvancedWeb\footballStat> python ./api.py
* Tip: There are .env or .flaskenv files present. Do "pip install python-dotenv" to use them.
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

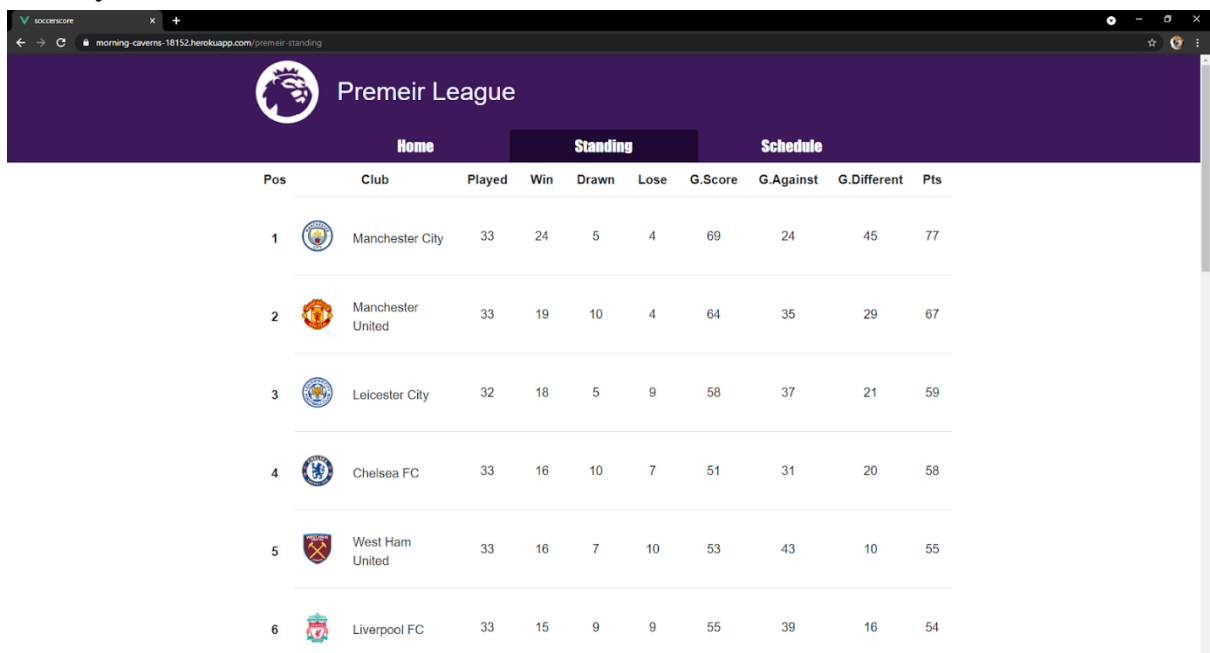
## โปรเจค SOCCERSCORE







URL : <https://morning-caverns-18152.herokuapp.com/>

1. หน้าแรกของโปรเจค SOCCERSCORE ก็จะมีให้เลือก 2 ลีก ก็คือ Premier League และ LaLiga

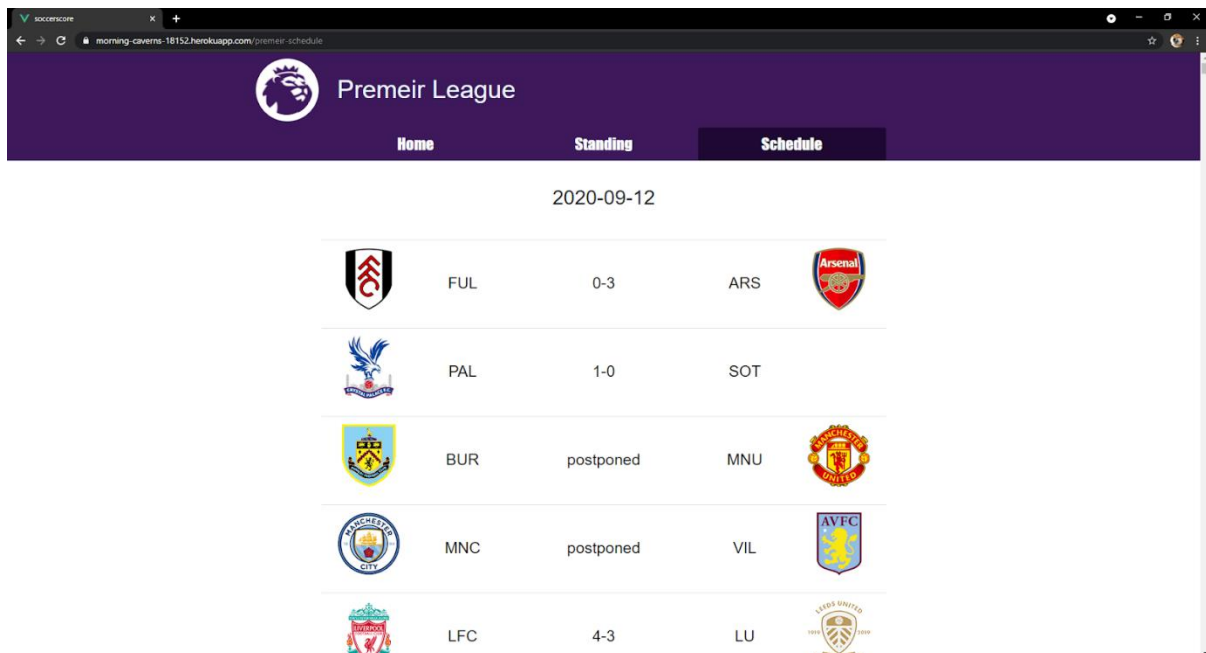











2. เมื่อคลิกเข้าไปในลีกที่เลือกเช่น Premier League หน้าแรกก็จะเป็น Standing บอกลำดับ 20 ลำดับในลีก และข้อมูลการแข่งขันของแต่ละทีม เช่น แพ้กี่ครั้ง ชนะกี่ครั้ง และบอกคะแนนรวมเป็นต้น

The image shows a web browser window displaying the Premier League Standing page. The page has a dark purple header with the Premier League logo and the text 'Premeir League' (note the typo). Below the header is a navigation bar with three tabs: 'Home', 'Standing' (which is selected and highlighted), and 'Schedule'. The main content area displays a table of the league standings. The table has columns for Position (Pos), Club, Played, Win, Drawn, Lose, Goals Scored (G.Score), Goals Against (G.Against), Goals Difference (G.Different), and Points (Pts). The table lists the top six teams: Manchester City, Manchester United, Leicester City, Chelsea FC, West Ham United, and Liverpool FC, along with their respective statistics.

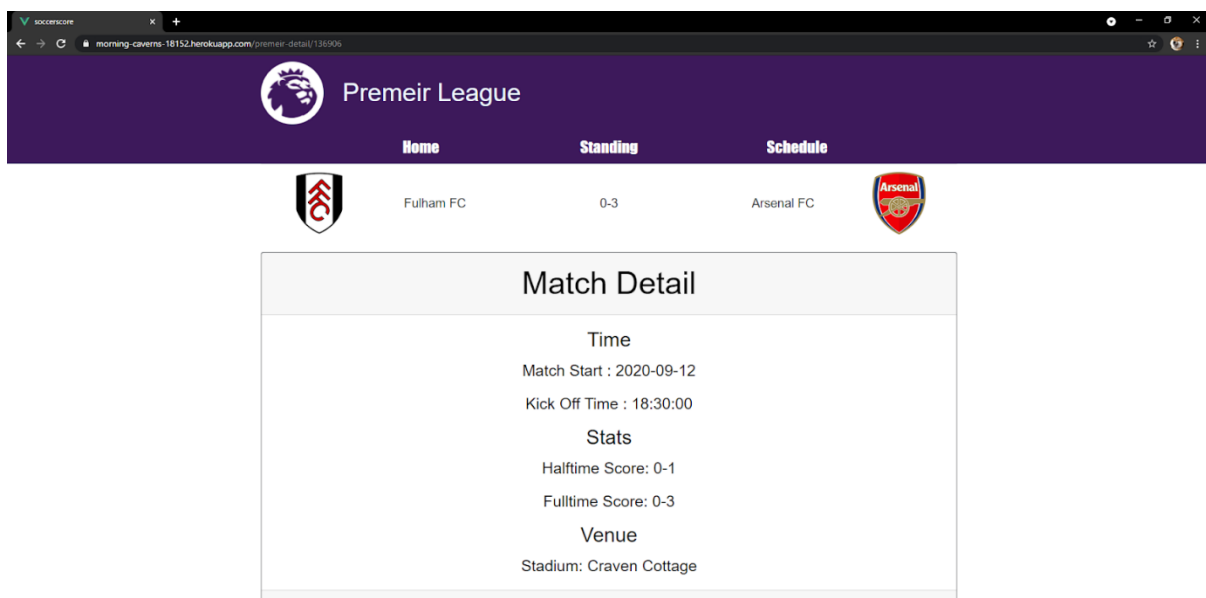
Pos	Club	Played	Win	Drawn	Lose	G.Score	G.Against	G.Different	Pts
1	 Manchester City	33	24	5	4	69	24	45	77
2	 Manchester United	33	19	10	4	64	35	29	67
3	 Leicester City	32	18	5	9	58	37	21	59
4	 Chelsea FC	33	16	10	7	51	31	20	58
5	 West Ham United	33	16	7	10	53	43	10	55
6	 Liverpool FC	33	15	9	9	55	39	16	54



3. หน้า Schedule เป็นการบอกตารางการแข่งขันในลีกว่าวันไหนทีมไหนจะเจอคู่แข่ง และสามารถเข้าไปคลิกดู Match Detail ของคู่ที่จะแข่งได้อีกด้วย



Premier League				
Home Standing Schedule				
2020-09-12				
	FUL	0-3	ARS	
	PAL	1-0	SOT	
	BUR	postponed	MNU	
	MNC	postponed	VIL	
	LFC	4-3	LU	

4. หน้า Match Detail บอกรายละเอียดเวลาการแข่งขันที่ คะแนนในแต่ละ Stats

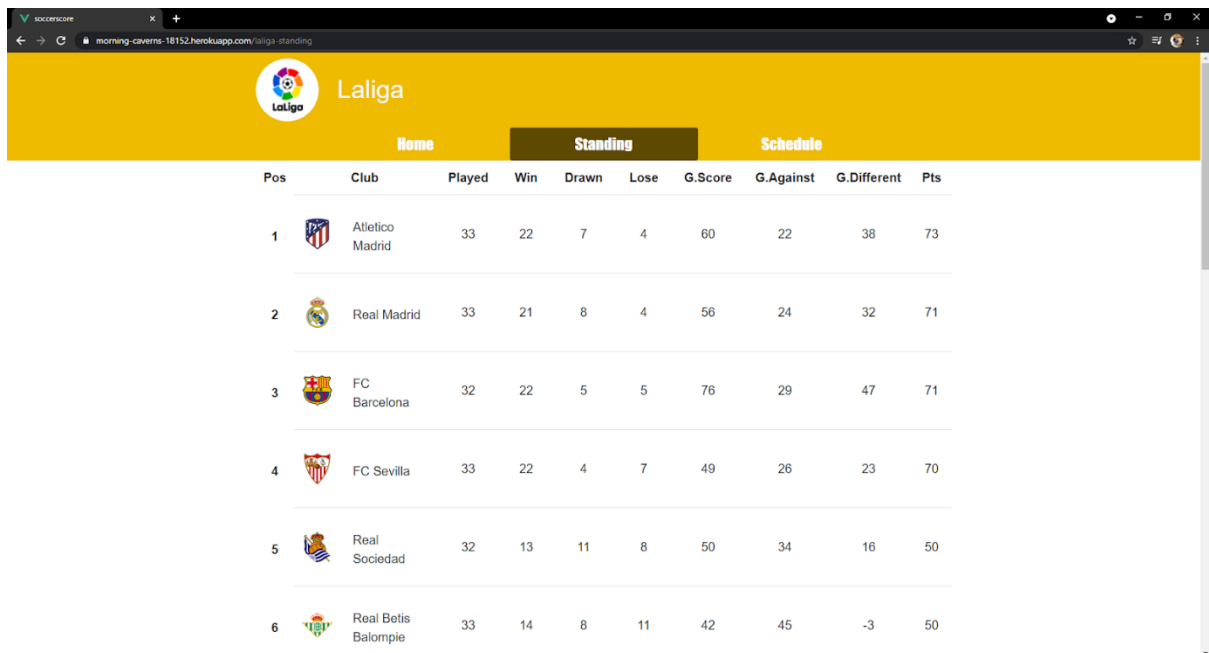








Premier League				
Home Standing Schedule				
	Fulham FC	0-3	Arsenal FC	
Match Detail				
Time				
Match Start : 2020-09-12				
Kick Off Time : 18:30:00				
Stats				
Halftime Score: 0-1				
Fulltime Score: 0-3				
Venue				
Stadium: Craven Cottage				

5. ถ้าต้องการกลับไปหน้าแรกให้คลิกที่ Home

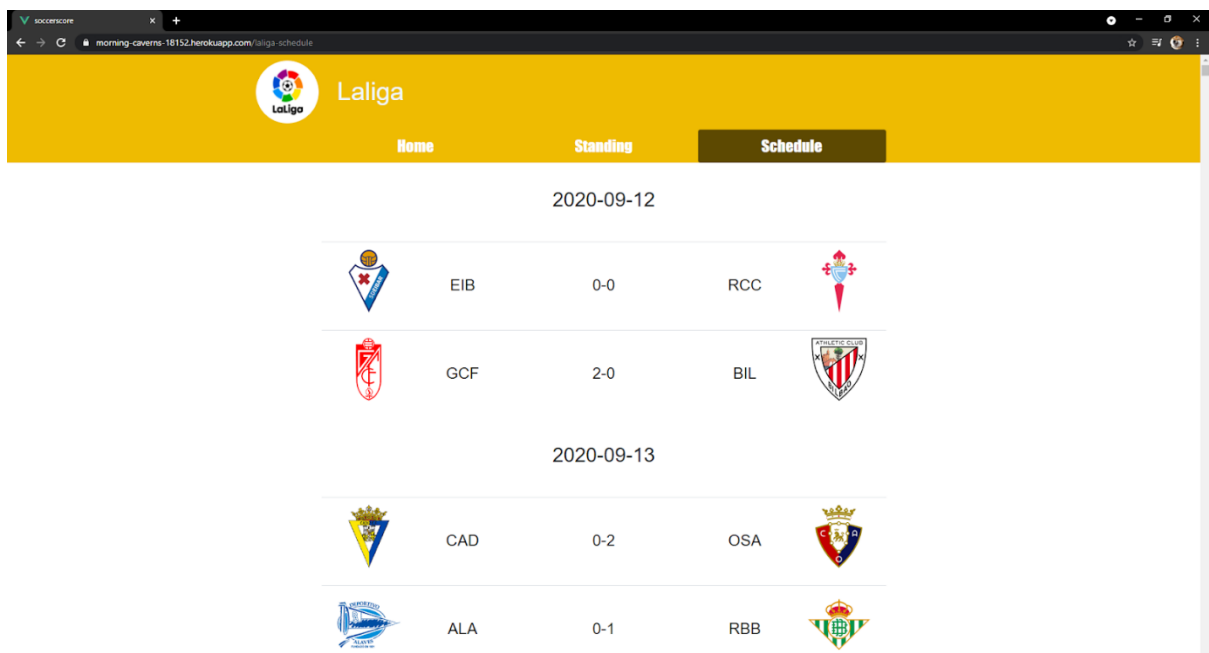
## ตัวอย่างหน้าเว็บที่แสดงลีก Laliga









### 1.หน้า Standing



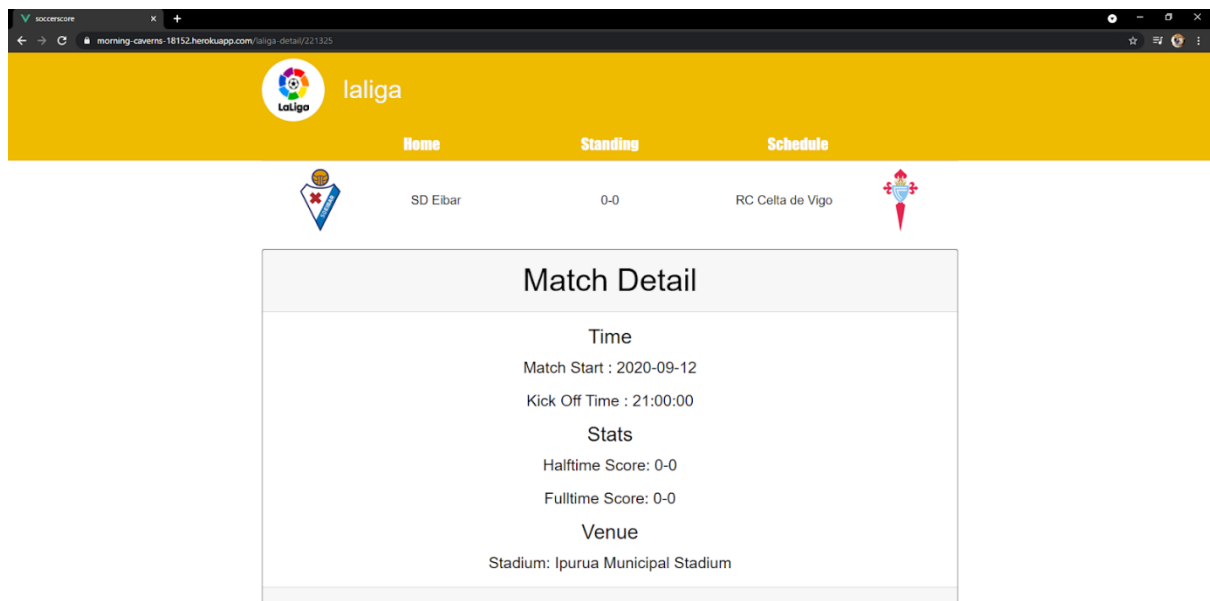
Pos	Club	Played	Win	Drawn	Lose	G.Score	G.Against	G.Different	Pts
1	 Atletico Madrid	33	22	7	4	60	22	38	73
2	 Real Madrid	33	21	8	4	56	24	32	71
3	 FC Barcelona	32	22	5	5	76	29	47	71
4	 FC Sevilla	33	22	4	7	49	26	23	70
5	 Real Sociedad	32	13	11	8	50	34	16	50
6	 Real Betis Balompie	33	14	8	11	42	45	-3	50

### 2.หน้า Schedule



2020-09-12				
	EIB	0-0	RCC	
	GCF	2-0	BIL	
2020-09-13				
	CAD	0-2	OSA	
	ALA	0-1	RBB	

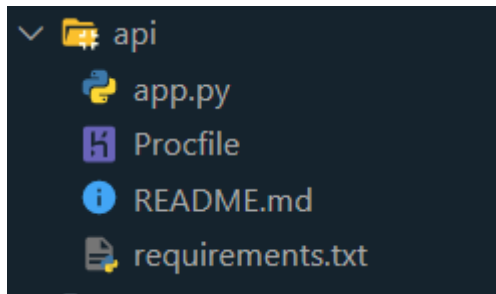
### 3.หน้า Match Detail





## CODE

### ส่วน Code ของ API



app.py คือไฟล์ที่ทำการจัดการฐานข้อมูล และเป็น API สำหรับการส่งข้อมูลไปแสดงผลยังหน้าเว็บ

```
from flask import Flask, request
from flask_cors import CORS
from pymongo import MongoClient
from bson import json_util
import requests
from datetime import datetime
from pytz import timezone
from apscheduler.schedulers.background import BackgroundScheduler

app = Flask(__name__)
cors = CORS(app, resources={r"/*": {"origins": "*"}})

# mongo database URI
client = MongoClient(
    "mongodb+srv://admin:3cMr8dQtNeKzROXq@cluster0.hgtii.mongodb.net"
    # "mongodb://admin:NNApnn47420@node13003-soccer-api.app.ruk-com.cloud:11027"
)

# database name
db = client["SoccerScore"]
Leagues = db.Leagues
Teams = db.Teams
Matches = db.Matches
Standings = db.Standings
headers = {"apikey": "6f889a60-8edd-11eb-9084-05c23de9546d"}
```

```
@app.route("/leagues/", methods=["GET"])
```

```
def getAllLeague():
```

```
    data = Leagues.find({})
```

```
    return json_util.dumps({"data": data})
```

```
@app.route("/standings/<ss_id>", methods=["GET"])
```

```
def getLeagueStandings(ss_id):
```

```
    # current premierLeague ss = 352 and LaLiga = 1511
```

```
    seasons_id = int(ss_id)
```

```
    cursors = Standings.find({"season_id": seasons_id})
```

```
    data = list()
```

```
    # setting format the json data
```

```
    for standing_obj in cursors:
```

```
        standings = standing_obj["standings"]
```

```
        # print(type(standings))
```

```
        # find match team_data (team_name and team_logo) and put to array data
```

```
        for team in standings:
```

```
            team_id = team["team_id"]
```

```
            # find team_data
```

```
            team_data = Teams.find({"team_id": team_id})
```

```
            for item in team_data:
```

```
                team["team_logo"] = item["logo"]
```

```
                team["team_name"] = item["name"]
```

```
    data = standings
```

```
    return json_util.dumps({"data": data})
```

```

@app.route("/matches/<ss_id>", methods=["GET"])
def getMatches(ss_id):
    # current premierLeague ss = 352 and LaLiga = 1511
    seasons_id = int(ss_id)

    # query string (date_from, date_to)
    args = request.args

    # check date filter is exist
    # date format is Y-m=d H:M:S
    if "date_from" in args:
        date_from = args["date_from"]
        # print("from", date_from)

    if "date_to" in args:
        date_to = args["date_to"]
        # print("to", date_to)

    # if use all filter
    data = Matches.find({"season_id": seasons_id, "match_start_th": {"$gt": date_from,
"$lt": date_to}})
    return json_util.dumps({"data": data})

    # if use only date_from
    data = Matches.find({"season_id": seasons_id, "match_start_th": {"$gt": date_from}})

else:
    # not use filter
    data = Matches.find({"season_id": seasons_id})

    return json_util.dumps({"data": data})

@app.route("/match-detail/<m_id>")

```

```
def getMatchDetail(m_id):
    match_id = int(m_id)
    data = Matches.find_one({"match_id": match_id})
    return json_util.dumps({"data": data})
```

```
def prepareDB():
    # print("start update Database")
    prepareLeaguesDB()
    prepareTeamDB()
    prepareMatchesDB()
    prepareStandings()
    # print("Database update was successful. ")
    return 0
```

```
#### LeaguesDB ####
```

```
# find league_id, season_id and country_id in subscribLeagues
```

```
def prepareLeaguesDB():
    leagues = getSubscribLeagues()
    seasons = getLeagueSeasonInfo(leagues)

    for i in range(len(leagues)):
        # filter before put to db
        league_data = dict()
        league_data = leagues[i]
        league_data["season_data"] = seasons[i]

        filter = {"league_id": league_data["league_id"]}
        update = {"$set": league_data}
        Leagues.update_one(filter=filter, update=update, upsert=True)
```

```
def getSubscribLeagues():
    params = {"subscribed": True}
```

```

    response = requests.get("https://app.sportdataapi.com/api/v1/soccer/leagues",
headers=headers, params=params).json()
    datas = response["data"]

    return datas

```

```

def getLeagueSeasonInfo(leagues):

```

```

    seasonData = list()
    for league in leagues:
        params = {"league_id": league["league_id"]}
        response = requests.get(
            "https://app.sportdataapi.com/api/v1/soccer/seasons", headers=headers,
params=params
        ).json()
        seasonData.append(response["data"])

    return seasonData

```

```

### TeamDB ###

```

```

# find every team in country is subscrib

```

```

def prepareTeamDB():

```

```

    for league in Leagues.find({}):
        # find team by country_id
        params = {"country_id": league["country_id"]}
        response = requests.get(
            "https://app.sportdataapi.com/api/v1/soccer/teams", headers=headers,
params=params
        ).json()

        # update team_data and insert team isn't exist

```

```

team_data = response["data"]
for data in team_data:
    filter = {"team_id": data["team_id"]}
    update = {"$set": data}
    Teams.update_one(filter=filter, update=update, upsert=True)

#### MatchesDB ####
def prepareMatchesDB():

    current_season_list = getAllCurrentSeasonID() # get current season from subscrib Leagues
    url = "https://app.sportdataapi.com/api/v1/soccer/matches"

    datas = list()
    for i in range(len(current_season_list)):
        current_season = current_season_list[i]
        params = {"season_id": current_season}
        datas.append(requests.get(url, headers=headers, params=params).json()["data"])

    # sort matches data orderby date
    sortedData = sorted(datas[i], key=lambda data: data["match_start_iso"])

    for match in sortedData:
        filter = {"match_id": match["match_id"]}
        match["match_start_th"] = toThaiTime(match["match_start"])
        match["match_date_th"] = match["match_start_th"][0:10]
        match["match_time_th"] = match["match_start_th"][11:]
        update = {"$set": match}
        Matches.update_one(filter=filter, update=update, upsert=True)

def getAllCurrentSeasonID():
    current_season_list = list()

```

```

for obj in Leagues.find({}):
    seasons = obj["season_data"]
    for season in seasons:
        if season["is_current"] == 1:
            current_season_list.append(season["season_id"])
            break
return current_season_list

```

```

def toThaiTime(str_time):
    # time format
    fmt = "%Y-%m-%d %H:%M:%S"

    # convert str to datetime
    time_obj = datetime.strptime(str_time, fmt)

    # set default time zone
    timezone_default = timezone("UTC")
    default_datetime_obj = timezone_default.localize(time_obj)

    # convert thai time zone
    timezone_bkk = timezone("Asia/Bangkok")
    bkk_datetime_obj = default_datetime_obj.astimezone(timezone_bkk)

    return bkk_datetime_obj.strftime(fmt)

```

### Standings Ranking ###

```

def prepareStandings():

    current_season_list = getAllCurrentSeasonID()
    url = "https://app.sportdataapi.com/api/v1/soccer/standings"

```

```


for i in range(len(current_season_list)):
    current_season = current_season_list[i]
    params = {"season_id": current_season}
    data = requests.get(url, headers=headers, params=params).json()["data"]

    filter = {"season_id": data["season_id"]}
    update = {"$set": data}
    Standings.update_one(filter=filter, update=update, upsert=True)

if __name__ == "__main__":
    update_schedule = BackgroundScheduler(daemon=True)
    update_schedule.add_job(prepareDB, "interval", hours=60)
    update_schedule.start()
    prepareDB()
    app.run(debug=True, port=80)

```

## Procfile คือ Configuration File ของ Heroku



```

Procfile
1 web: gunicorn app:app
2

```

requirements.txt คือรายชื่อของ packet ที่จำเป็นต้องใช้ในการทำงาน

Flask==1.1.2

pymongo[srv]

Flask\_Cors==3.0.10

pytz==2021.1

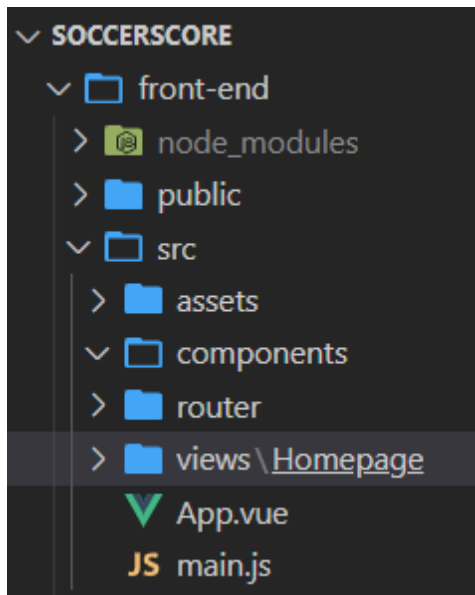
requests==2.25.1

apscheduler

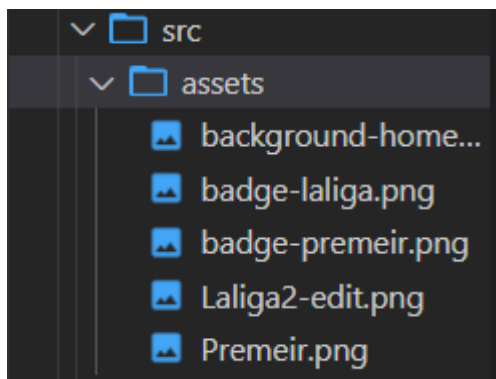
gunicorn

## ส่วน Code ของ Vue

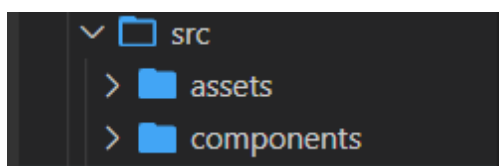




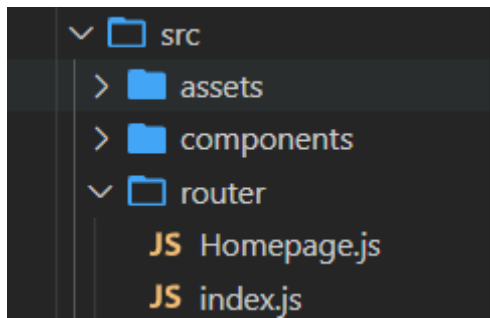
หน้าส่วนแสดงโครงสร้างทั้งหมดของโปรเจ็คโดยจะมีอยู่ 2 โฟลเดอร์ใหญ่และ 4 โฟลเดอร์แยก



ในโฟลเดอร์ย่อยของ src อันแรก จะเป็นโฟลเดอร์ assets ซึ่งเอาไว้เก็บรูปภาพต่างๆที่ใช้ในการแสดงผลของโปรแกรมทั้งหมด



ในส่วนของโฟลเดอร์ที่ 2 จะชื่อ components โดยจะเอาไว้จัดเก็บ components ต่างๆที่ได้สร้างขึ้นมา



โพลเดอร์ที่ 3 เป็นโพลเดอร์ router ที่จะเอาไว้ใส่ลิงค์ของ components ต่างๆ Path ต่างๆ ของ components

### ไฟล์ Homepage.js

```
export default {  
  path: '/',  
  component: () => import('@views/Homepage/index.vue'),  
  children: [  
    {  
      path: '',  
      name: 'Detail',  
      component: () => import('@views/Homepage/pages/SoccerStadium.vue')  
    },  
    {  
      path: 'premeir-standing',  
      name: 'FirstMenu',  
      component: () => import('@views/Homepage/pages/premeir-standing.vue')  
    },  
    {  
      path: 'premeir-schedule',  
      name: 'premeir-schedule',  
      component: () => import('@views/Homepage/pages/premeir-schedule.vue')  
    },  
    {  
      path: 'premeir-detail/:MatchDetail',  
      name: 'premeir-detail',  
      component: () => import('@views/Homepage/pages/premeir-detail.vue'),  
    }  
  ]  
}
```

```

      props: true
    },
    {
      path: 'laliga-schedule',
      name: 'laliga-schedule',
      component: () => import('@/views/Homepage/pages/laliga-schedule.vue')
    },
    {
      path: 'laliga-standing',
      name: 'laliga-standing',
      component: () => import('@/views/Homepage/pages/laliga-standing.vue')
    },
    {
      path: 'laliga-detail/:MatchDetail',
      name: 'laliga-detail',
      component: () => import('@/views/Homepage/pages/laliga-detail.vue'),
      props: true
    }
  ]
}

```

### ไฟล์ index.js

```

import Vue from 'vue'
import VueRouter from 'vue-router'
import Homepage from './Homepage'
Vue.use(VueRouter)

```

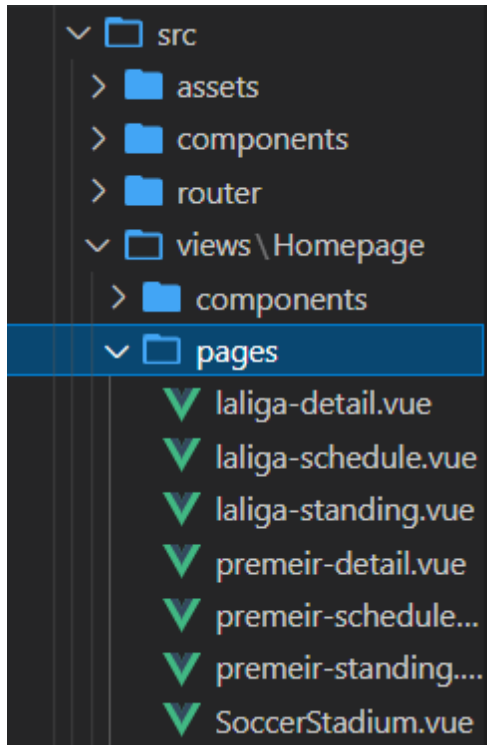
```

const routes = [
  Homepage
]
const router = new VueRouter({
  mode: 'history',
  routes

```

})

export default router



ในโฟลเดอร์ที่ 4 ชื่อ views เป็นโฟลเดอร์ที่ส่วนตัวเองไว้เก็บ components ที่ใช้ โดยมี 2 โฟลเดอร์ย่อย โดยมี components และ pages โดย pages จะเอาไว้เก็บ components ทุกตัวที่ใช้ในหน้าต่างๆ

### ไฟล์ laliga-detail.vue

```
<template>
  <div>
    <!-- title img logo and Premeir league text-->
    <div class="background-titlebar">
      <div class="container">
        <div class="image">
          
        <h1>laliga</h1>
    </div>
</div>
<!-- end title img logo and Premeir league text-->

<!-- nav menu bar -->
<div class="navbar">
    <ul class="nav nav-pills nav-fill red">
        <li class="nav-item">
            <a class="nav-link " href="/">Home</a>
        </li>
        <li class="nav-item ">
            <a class="nav-link" href="/laliga-standing">Standing</a>
        </li>
        <li class="nav-item ">
            <a class="nav-link" href="/laliga-schedule">Schedule</a>
        </li>
    </ul>
</div>
<!-- end of nav menu bar -->

</div>
<!-- end of title -->

<!-- body & table -->
<div class="container">

    <!-- match show -->
    <!-- ทำการวน loop for เพื่อดึงข้อมูลมาจาก object -->
    <div class="table table-hover table-detail">
        <table
            v-for="team in info.data"

```

```

:key="team"
class="list table table-striped "
>
<!-- กรณีที่ Match มีการเลื่อนออกไปจะไม่โชว์ score -->
<tr v-if="team.status != 'postponed'">
  <td width="150px"></td>
  <td width="150px">{{ team.home_team.name }}</td>
  <td width="300px">{{ team.stats.ft_score }}</td>
  <td width="150px">{{ team.away_team.name }}</td>
  <td width="150px"></td>
</tr>
</table>
</div>
<!-- end match show -->

<!-- card show stat and detail -->
<!-- ใช้ card เพื่อสร้างเป็นหน้าต่างเพื่อให้ดูสวยงาม -->
<div class="description">
  <div class="card text-center border-dark mb-3">
    <div class="card-header">
      Match Detail
    </div>
    <div class="card-body" v-for="team in info.data" :key="team">
      <h2 class="card-title">Time</h2>
      <p class="card-text">Match Start : {{ team.match_date_th }}</p>
      <p class="card-text">Kick Off Time : {{ team.match_time_th }}</p>
      <h2 class="card-title">Stats</h2>
      <p class="card-text">Halftime Score: {{ team.stats.ht_score }}</p>
      <p class="card-text">Fulltime Score: {{ team.stats.ft_score }}</p>
      <h2 class="card-title">Venue</h2>
      <p class="card-text">Stadium: {{ team.venue.name }}</p>
    </div>
    <div class="card-footer text-muted"></div>
  </div>
</div>

```

```

        </div>
    </div>

    <!-- end card show stat and detail -->

</div>
<!-- end body & table -->
</div>
</template>

<script>
import axios from 'axios'
export default {
  data () {
    return {
      // ตัวแปรที่นำมาเก็บค่าจาก api
      info: {}
    }
  },
  // get match_id by props
  // ใช้ function props เพื่อนำค่า params ที่ได้ส่งมาจากหน้า schedule มาใช้
  props: ['MatchDetail'],
  // get api function
  mounted () {
    // รับค่าจาก api โดยใช้ axios
    axios
    // นำตัวแปรที่ได้มาจาก props มาต่อท้ายเข้า api จะได้ข้อมูลเฉพาะของแต่ละ Match
    .get('https://soccerscoreapi.herokuapp.com/match-detail/' + this.MatchDetail)
    .then(response => {
      this.info = response
    })
  }
}
</script>

```

```
<style scoped>
```

```
.table {  
  font-size: 20px;  
  font-family: "B612", sans-serif;  
}
```

```
a {  
  font-size: 25px;  
  color: white;  
}
```

```
a:hover {  
  background-color: #5d4901;  
}
```

```
p {  
  font-size: 25px;  
}
```

```
td,  
th {  
  text-align: center;  
  vertical-align: middle;  
}
```

```
.nav-link {  
  width: 300px;  
  margin: 0;  
}
```

```
.description {  
  font-size: 50px;  
  color: black;
```



```
font-family: sans-serif;
}

.nav-pills .active .nav-link:not(.active) {
  background-color: #5d4901;
}

.navbar {
  margin-top: 2px;
  display: flex;
  align-items: center;
  justify-content: center;
  padding-bottom: 0px;
}

.team-logo {
  width: 30px;
  height: 30px;
}

.tab-bar {
  display: flex;
  align-items: center;
  justify-content: center;
  color: black;
  padding-left: 10px;
  padding-right: 10px;
  margin-top: 10px;
}

.container {
  width: 90vw;
  margin: 0 auto;
}

.background-titlebar {
  width: 100%;
  height: fit-content;
```

```
background-color: #eebb01;
}
.image {
  display: flex;
  align-items: center;
  font-family: Regular, sans-serif, Georgia;
}
.image-position {
  background-color: white;
  padding: 10px;
  padding-left: 10px ;
  width: 80px;
  height: 80px;
  background: white;
  -moz-border-radius: 70px;
  -webkit-border-radius: 70px;
  border-radius: 70px;
  margin-top: 10px;
}

h1 {
  margin-left: 30px;
  color: aliceblue;
  margin-top: 15px;
}
</style>
```

ไฟล์ `laliga-schedule.vue`

```
<template>
  <div>

    <!-- title of page -->
```

```
<div class="background">
```

```
<!-- title img logo and Premeir league text-->
```

```
<div class="container">
```

```
<div class="image">
```

```

```

```
<h1>Laliga</h1>
```

```
</div>
```

```
</div>
```

```
<!-- end title img logo and Premeir league text-->
```

```
<!-- nav menu bar -->
```

```
<div class="navbar">
```

```
<ul class="nav nav-pills nav-fill red">
```

```
<li class="menu nav-item">
```

```
<a class="nav-link " aria-current="page" href="/">Home</a>
```

```
</li>
```

```
<li class="menu nav-item ">
```

```
<a class="nav-link" href="laliga-standing">Standing</a>
```

```
</li>
```

```
<li class="menu nav-item active">
```

```
<a class="nav-link" href="laliga-schedule">Schedule</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
<!-- end nav menu bar -->
```

```
</div>
```

```
<!-- end title of page -->
```

```

<!-- body & table -->
<div class="container">

  <!-- for date of the match -->
  <!-- นำวันที่ในแต่ละ Match ออกมาแสดงโดยสร้างตัวแปรรับค่า object โดยใช้ loop for -->
  <div class="table" v-for="date in NewDate" :key="date">
    <h2>{{ date }}</h2>
  <!-- end for date of the match -->

  <!-- data in table -->
  <!-- ทำการวน loop เพื่อดึงค่าข้อมูล Match ต่างๆมาแสดง -->
  <div v-for="team in info.data.data" :key="team">
    <table v-if="date == team.match_date_th" class="table table-hover">
      <tbody>

        <!-- send match id to match-detail -->
        <!-- ใช้ router link เพื่อเป็นการส่ง params ไปยังหน้า Match Detail เพื่อ Filler เฉพาะ
Match นั้นๆ -->
        <router-link
          :to="{
            name: 'laliga-detail',
            params: { MatchDetail: team.match_id }
          }"
          class="match-detail"
        >

        <!-- end send match id to match-detail -->

        <!-- data in table -->
        <tr>
          <td width="150px">
            
          </td>
          <td width="150px" class="inrow">

```

```
    {{ team.home_team.short_code }}  
</td>
```

```
<!-- if statement -->
```

<!-- ในส่วนของ row นี้จะเป็นการนำสกอร์ของแต่ละ Match ออกมาแต่เนื่องจากมีบาง Match ที่เลื่อนหรือยังไม่ได้แข่งจึงใช้ if เช็คเงื่อนไขก่อนแสดงผล -->

```
<!-- กรณีแข่งขันจบแล้ว -->
```

```
<td  
    width="300px"  
    class="inrow"  
    v-if="team.status == 'finished'"  
>  
    {{ team.stats.ft_score }}  
</td>
```

```
<!-- กรณียังไม่ได้แข่งขัน -->
```

```
<td  
    width="300px"  
    class="inrow"  
    v-if="team.status == 'notstarted'"  
>  
    {{ team.match_time_th }}  
</td>
```

```
<!-- กรณีกำลังแข่งขันอยู่หรือเลื่อนการแข่งขันออกไป -->
```

```
<td  
    width="300px"  
    class="inrow"  
    v-if="team.status == 'postponed' || team.status == 'inplay'"  
>  
    {{ team.status }}  
</td>
```

```
<!-- กรณียังไม่ได้มีการประกาศเวลาแข่งขันออกมา -->
```

```
<td
```

```
width="300px"
```

```
class="inrow"
```

```
v-if="team.status_code == 17"
```

```
>
```

```
TBA
```

```
</td>
```

```
<!-- end if statement -->
```

```
<td width="150px" class="inrow">
```

```
{{ team.away_team.short_code }}
```

```
</td>
```

```
<td width="150px">
```

```

```

```
</td>
```

```
</tr>
```

```
<!-- end of data in tale -->
```

```
</router-link>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</template>
```

```
<script>
```

```
import axios from 'axios'
```

```
export default {
```

```
data () {
```

```
return {
```

```

// ตัวแปรที่นำมาเก็บค่าจาก api
info: {},
// ตัวแปรที่นำมาเก็บค่าจาก api
date: {},
// ตัวแปรที่สร้างมาไว้สำหรับสร้าง set เพื่อบันทึกวันที่ ที่เหมือนกันออกเหลือ 1 วัน
arrayMap: null,
// นำมาเก็บค่าจาก arrayMap อีกที
NewDate: null

// JSON.stringify(jsArray) converts the jsArray into a string which can be stored in
sessionStorage
}
},
// get api function
mounted () {
// รับค่าจาก api โดยใช้ axios
axios
.get(
'https://soccerscoreapi.herokuapp.com/matches/1511'
)
.then(response => {
// รับค่าจากทั้งหมดจาก api
this.info = response
// รับเฉพาะ object ที่มีวันอยู่
this.date = response.data.data
// ทำการ map date เข้ากับ ค่าข้างใน object ที่มีวันอยู่ โดยเก็บไว้ที่ arrayMap
this.arrayMap = this.date.map(date => date.match_date_th)
// สร้างตัวแปรมารับค่าที่นำไปเข้า set แล้วจะได้วันที่ออกมา
this.NewDate = new Set(this.arrayMap)

})
}
}

```

```
</script>
```

```
<style scoped>
```

```
.table {  
  font-size: 20px;  
  font-family: "B612", sans-serif;  
}
```

```
.table {  
  text-align: center;  
  vertical-align: middle;  
}
```

```
a {  
  font-size: 25px;  
  color: white;  
}
```

```
h2 {  
  padding: 40px;  
  font-size: 40px bold;  
}
```

```
.match-detail {  
  color: black;  
  text-decoration: none;  
}
```

```
.inrow {  
  padding-top: 50px;  
}
```

```
.menu a:hover {
```



```
background-color: #5d4901;
text-decoration: none;
}
```

```
.nav-link {
width: 300px;
margin: 0;
}
```

```
.nav-pills .active .nav-link:not(.active) {
background-color: #5d4901;
}
```

```
.navbar {
margin-top: 2px;
display: flex;
align-items: center;
justify-content: center;
padding-bottom: 0px;
}
```

```
.match-detail {
align-content: center;
}
```

```
.container {
width: 90vw;
margin: 0 auto;
}
```

```
.container table {
display: flex;
justify-content: center;
}
```

```
.background {
  width: 100%;
  height: fit-content;
  background-color: #eebb01;
}

.image {
  display: flex;
  align-items: center;
  font-family: Regular, sans-serif, Georgia;
}

.image-position {
  background-color: white;
  padding: 10px;
  padding-left: 10px ;
  width: 80px;
  height: 80px;
  background: white;
  -moz-border-radius: 70px;
  -webkit-border-radius: 70px;
  border-radius: 70px;
  margin-top: 10px;
}

h1 {
  margin-left: 30px;
  color: aliceblue;
  margin-top: 15px;
}
```

</style>

ไฟล์ **laliga-standing.vue**

<template>

<div>

<!-- title -->

<div class="background-titlebar">

```
<!-- title img logo and Premeir league text-->
<div class="container">
  <div class="image">
    <!-- ทำการเพิ่ม logo ของ League โดยใช้ <img> -->
    
    <h1>Laliga </h1>
  </div>
</div>
<!-- end title img logo and Premeir league text-->
```

```
<!-- nav menu bar -->
<div class="navbar">
  <!-- ทำเป็นเมนูสำหรับคลิกเลือก page โดยใช้ navbar -->
  <ul class="nav nav-pills nav-fill red">
    <li class="nav-item">
      <a class="nav-link " aria-current="page" href="/">Home</a>
    </li>
    <!-- เมื่อเราอยู่ตำแหน่งหน้าไหนจะให้ปุ่ม active ที่เมนูนั้นๆ -->
    <li class="nav-item active">
      <a class="nav-link" href="laliga-standing">Standing</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="laliga-schedule">Schedule</a>
    </li>
  </ul>
</div>
<!-- end of nav menu bar -->
```

```
</div>
```

```
<!-- end of title -->
```

```
<!-- body & table -->
```

```
<div class="container">
```

```
<!-- table -->
```

```
<div class="table-data borderless">
```

```
<!-- head table & columns -->
```

```
<!-- สร้างหัวตารางตามที่ได้ออกแบบไว้ -->
```

```
<table class="table ">
```

```
<thead class="">
```

```
<tr>
```

```
<th scope="col">Pos</th>
```

```
<th scope="col" colspan="2">Club</th>
```

```
<th scope="col">Played</th>
```

```
<th scope="col">Win</th>
```

```
<th scope="col">Drawn</th>
```

```
<th scope="col">Lose</th>
```

```
<th scope="col">G.Score</th>
```

```
<th scope="col">G.Against</th>
```

```
<th scope="col">G.Different</th>
```

```
<th scope="col">Pts</th>
```

```
</tr>
```

```
</thead>
```

```
<!-- end head table & columns -->
```

```
<!-- body data standing -->
```

```
<tbody>
```

```
<!-- วง loop เพื่อดึงค่าออกจากตัวแปร object ที่ได้ get ค่าจาก api มา -->
```

```
<tr v-for="item in info.data.data" :key="item">
```

```
<th scope="row" class="pos-position">{{ item.position }}</th>
```

```
<td class="team-image">
```

```

        
      </td>
      <td width="300px" class="team-name">{{ item.team_name }}</td>
      <td width="180px" class="">{{ item.overall.games_played }}</td>
      <td width="180px">{{ item.overall.won }}</td>
      <td width="180px">{{ item.overall.draw }}</td>
      <td width="180px">{{ item.overall.lost }}</td>
      <td width="180px">{{ item.overall.goals_scored }}</td>
      <td width="180px">{{ item.overall.goals_against }}</td>
      <td width="180px">{{ item.overall.goals_diff }}</td>
      <td width="100">{{ item.points }}</td>
    </tr>
  </tbody>
<!-- end body data standing -->

</table>
</div>
<!-- end table -->

</div>
<!-- end body & table -->
</div>
</template>

<script>
import axios from 'axios'
export default {
  data () {

```

```

    return {
      info: {}

      // JSON.stringify(jsArray) converts the jsArray into a string which can be stored in
      sessionStorage

    }
  },
  mounted () {
    axios.get('https://soccerscoreapi.herokuapp.com/standings/1511').then(response => {
      this.info = response
    })
  }
}
</script>

```

```

<style scoped>
.table {
  font-size: 20px;
  font-family: "B612", sans-serif;
}

```

```

.borderless th {
  border: none;
}
a {
  font-size: 25px;
  color: white;
}
a:hover {
  background-color: #5d4901;
}

```

```

td , th{

```

```
text-align: center;
vertical-align: middle;
}
```

```
.nav-link {
width: 300px;
margin: 0;
}
```

```
.nav-pills .active .nav-link:not(.active) {
background-color: #5d4901;
}
```

```
.navbar {
margin-top: 2px;
display: flex;
align-items: center;
justify-content: center;
padding-bottom: 0px;
}
```

```
.pos-position {
padding-top: 22px;
}
```

```
.team-image {
display: flex;
align-items: left;
justify-content: left;
padding-top: 50%;
padding-bottom: 50%;
}
```

```
.team-name {
padding-top: 20px;
padding-left: 20px;
```

```
    text-align: left;
}
.team-logo {
    width: 30px;
    height: 30px;
}
.tab-bar {
    display: flex;
    align-items: center;
    justify-content: center;
    color: black;
    padding-left: 10px;
    padding-right: 10px;
    margin-top: 10px;
}
.container {
    width: 90vw;
    margin: 0 auto;
}
.background-titlebar {
    width: 100%;
    height: fit-content;
    background-color: #eebb01;
}
.image {
    display: flex;
    align-items: center;
    font-family: Regular, sans-serif, Georgia;
}
.image-position {
    background-color: white;
    padding: 10px;
    padding-left: 10px ;
```



```
width: 80px;
height: 80px;
background: white;
-moz-border-radius: 70px;
-webkit-border-radius: 70px;
border-radius: 70px;
margin-top: 10px;
}
```

```
h1 {
  margin-left: 30px;
  color: aliceblue;
  margin-top: 15px;
}
```

```
</style>
```

**ไฟล์ premeir-detail.vue**

```
<template>
  <div>
    <!-- title img logo and Premeir league text-->
    <div class="background-titlebar">
      <div class="container">
        <div class="image">
          
          <h1>Premeir League</h1>
        </div>
      </div>
    </div>
    <!-- end title img logo and Premeir league text-->

    <!-- nav menu bar -->
```

```

<div class="navbar">
  <ul class="nav nav-pills nav-fill red">
    <li class="nav-item">
      <a class="nav-link " href="/">Home</a>
    </li>
    <li class="nav-item ">
      <a class="nav-link" href="/premeir-standing">Standing</a>
    </li>
    <li class="nav-item ">
      <a class="nav-link" href="/premeir-schedule">Schedule</a>
    </li>
  </ul>
</div>

<!-- end of nav menu bar -->
</div>

<!-- end of title -->

<!-- body & table -->
<div class="container">

  <!-- match show -->
  <div class="table table-hover table-detail">\
    <!-- ทำการวน loop for เพื่อดึงข้อมูลมาจาก object -->
    <table
      v-for="team in info.data"
      :key="team"
      class="list table table-striped "
    >
      <!-- กรณีที่ Match มีการเลื่อนออกไปจะไม่โชว์ score -->
      <tr v-if="team.status != 'postponed'">
        <td width="150px"></td>
        <td width="150px">{{ team.home_team.name }}</td>
        <td width="300px">{{ team.stats.ft_score }}</td>
      </tr>
    </table>
  </div>
</div>

```

```

        <td width="150px">{{ team.away_team.name }}</td>
        <td width="150px"></td>
    </tr>
</table>
</div>
<!-- end match show -->

<!-- card show stat and detail -->
<!-- ใช้ card เพื่อสร้างเป็นหน้าต่างเพื่อให้ดูสวยงาม -->
<div class="description">
    <div class="card text-center border-dark mb-3">
        <div class="card-header">
            Match Detail
        </div>
        <div class="card-body" v-for="team in info.data" :key="team">
            <h2 class="card-title">Time</h2>
            <p class="card-text">Match Start : {{ team.match_date_th }}</p>
            <p class="card-text">Kick Off Time : {{ team.match_time_th }}</p>
            <h2 class="card-title">Stats</h2>
            <p class="card-text">Halftime Score: {{ team.stats.ht_score }}</p>
            <p class="card-text">Fulltime Score: {{ team.stats.ft_score }}</p>
            <h2 class="card-title">Venue</h2>
            <p class="card-text">Stadium: {{ team.venue.name }}</p>
        </div>
        <div class="card-footer text-muted"></div>
    </div>
</div>
<!-- end card show stat and detail -->

</div>
<!-- end body & table -->

</div>

```

</template>

<script>

import axios from 'axios'

export default {

  data () {

    return {

      // ตัวแปรที่นำมาเก็บค่าจาก api

      info: {}

    }

  },

  // get match\_id by props

  // ใช้ function props เพื่อนำค่า params ที่ได้ส่งมาจากหน้า schedule มาใช้

  props: ['MatchDetail'],

  // get api function

  mounted () {

    // รับค่าจาก api โดยใช้ axios

    axios

    // นำตัวแปรที่ได้มาจาก props มาต่อท้ายเข้า api จะได้ข้อมูลเฉพาะของแต่ละ Match

    .get('https://soccerscoreapi.herokuapp.com/match-detail/' + this.MatchDetail)

    .then(response => {

      this.info = response

    })

  }

}

</script>

<style scoped>

  .table {

    font-size: 20px;

    font-family: "B612", sans-serif;

}

```
a {  
  font-size: 25px;  
  color: white;  
}  
a:hover {  
  background-color: #200835;  
}
```

```
p {  
  font-size: 25px;  
}
```

```
td,  
th {  
  text-align: center;  
  vertical-align: middle;  
}
```

```
.nav-link {  
  width: 300px;  
  margin: 0;  
}
```

```
.description {  
  font-size: 50px;  
  color: black;  
  font-family: sans-serif;  
}
```

```
.nav-pills .active .nav-link:not(.active) {  
  background-color: #200835;  
}
```

```
.navbar {
```

```
margin-top: 2px;
display: flex;
align-items: center;
justify-content: center;
padding-bottom: 0px;
}
.team-logo {
width: 30px;
height: 30px;
}
.tab-bar {
display: flex;
align-items: center;
justify-content: center;
color: black;
padding-left: 10px;
padding-right: 10px;
margin-top: 10px;
}
.container {
width: 90vw;
margin: 0 auto;
}
.background-titlebar {
width: 100%;
height: fit-content;
background-color: #3d195b;
}
.image {
display: flex;
align-items: center;
font-family: Regular, sans-serif, Georgia;
}
```

```
.image-position {
  background-color: white;
  padding: 10px 10px;
  width: 100px;
  height: 100px;
  background: white;
  -moz-border-radius: 70px;
  -webkit-border-radius: 70px;
  border-radius: 70px;
  margin-top: 10px;
}
```

```
h1 {
  margin-left: 30px;
  color: aliceblue;
  margin-top: 15px;
}
```

</style>

**ไฟล์ premeir-schedule.vue**

```
<template>
```

```
<div>
```

```
<!-- title of page -->
```

```
<div class="background">
```

```
<!-- title img logo and Premeir league text-->
```

```
<div class="container">
```

```
<div class="image">
```

```

    <h1>Premeir League</h1>
  </div>
</div>
<!-- end title img logo and Premeir league text-->

<!-- nav menu bar -->
<div class="navbar">
  <ul class="nav nav-pills nav-fill red">
    <li class="menu nav-item">
      <a class="nav-link " aria-current="page" href="/">Home</a>
    </li>
    <li class="menu nav-item ">
      <a class="nav-link" href="premeir-standing">Standing</a>
    </li>
    <li class="menu nav-item active">
      <a class="nav-link" href="premeir-schedule">Schedule</a>
    </li>
  </ul>
</div>
<!-- end nav menu bar -->

</div>
<!-- end title of page -->

<!-- body & table -->
<div class="container">

  <!-- for date of the match -->
  <!-- นำวันที่ในแต่ละ Match ออกมาแสดงโดยสร้างตัวแปรรับค่า object โดยใช้ loop for -->
  <div class="table" v-for="date in NewDate" :key="date">
    <h2>{{ date }}</h2>
  </div>
  <!-- end for date of the match -->

```



```

<!-- data in table -->
<!-- ทำการ ววน loop เพื่อดึงค่าข้อมูล Match ต่างๆมาแสดง -->
<div v-for="team in info.data.data" :key="team">
  <table v-if="date == team.match_date_th" class="table table-hover">
    <tbody>

      <!-- send match id to match-detail -->
      <!-- ใช้ router link เพื่อเป็นการส่ง params ไปยังหน้า Match Detail เพื่อ Filler เฉพาะ
Match นั้นๆ -->
      <router-link
        :to="{
          name: 'premeir-detail',
          params: { MatchDetail: team.match_id }
        }"
        class="match-detail"
      >

      <!-- end send match id to match-detail -->

      <!-- data in table -->
      <tr>
        <td width="150px">
          
        </td>
        <td width="150px" class="inrow">
          {{ team.home_team.short_code }}
        </td>

        <!-- if statement -->
        <!-- ในส่วนของ row นี้จะเป็นการนำสกอร์ของแต่ละ Match ออกมาแต่เนื่องจากมีบาง
Match ที่เลื่อนหรือยังไม่ได้แข่งจึงใช้ if เช็คเงื่อนไขก่อนแสดงผล -->
        <!-- กรณีแข่งขันจบแล้ว -->
        <td

```

```
width="300px"
class="inrow"
v-if="team.status == 'finished'"
>
  {{ team.stats.ft_score }}
</td>
```

```
<!-- กรณียังไม่ได้แข่งขัน -->
<td
  width="300px"
  class="inrow"
  v-if="team.status == 'notstarted'"
>
  {{ team.match_time_th }}
</td>
```

```
<!-- กรณีกำลังแข่งขันอยู่หรือเลื่อนการแข่งขันออกไป -->
<td
  width="300px"
  class="inrow"
  v-if="team.status == 'postponed' || team.status == 'inplay'"
>
  {{ team.status }}
</td>
```

```
<!-- กรณียังไม่ได้มีการประกาศเวลาแข่งขันออกมา -->
<td
  width="300px"
  class="inrow"
  v-if="team.status_code == 17"
>
  TBA
</td>
```

```
<!-- end if statement -->
```

```
<td width="150px" class="inrow">
  {{ team.away_team.short_code }}
</td>
```

```
<td width="150px">
  
</td>
```

```
</tr>
```

```
<!-- end of data in tale -->
```

```
</router-link>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
<!-- end of data in table -->
```

```
</div>
```

```
</div>
```

```
<!-- end of body & table -->
```

```
</div>
```

```
</template>
```

```
<script>
```

```
import axios from 'axios'
```

```
export default {
```

```
  data () {
```

```
    return {
```

```
      // ตัวแปรที่นำมาเก็บค่าจาก api
```

```
      info: {},
```

```
      // ตัวแปรที่นำมาเก็บค่าจาก api
```

```
      date: {},
```

```
      // ตัวแปรที่สร้างมาไว้สำหรับสร้าง set เพื่อกำหนดวันที่ ที่เหมือนกันออกเหลือ 1 วัน
```

```

    arrayMap: null,
    // นำมาเก็บค่าจาก arrayMap อีกที่
    NewDate: null

    // JSON.stringify(jsArray) converts the jsArray into a string which can be stored in
    sessionStorage

  },
  // get api function
  mounted () {
    // รับค่าจาก api โดยใช้ axios
    axios
      .get(
        'https://soccerscoreapi.herokuapp.com/matches/352'
      )
      .then(response => {
        // รับค่าจากทั้งหมดจาก api
        this.info = response
        // รับเฉพาะ object ที่มีวันอยู่
        this.date = response.data.data
        // ทำการ map date เข้ากับ ค่าข้างใน object ที่มีวันอยู่ โดยเก็บไว้ที่ arrayMap
        this.arrayMap = this.date.map(date => date.match_date_th)
        // สร้างตัวแปรมารับค่าที่นำไปเข้า set แล้วจะได้วันที่ออกมา
        this.NewDate = new Set(this.arrayMap)
      })
  }
}
</script>

```

```

<style scoped>
.table {
  font-size: 20px;
  font-family: "B612", sans-serif;

```

```
}
```

```
/* set data in table to center */
```

```
.table {
```

```
    text-align: center;
```

```
    vertical-align: middle;
```

```
}
```

```
a {
```

```
    font-size: 25px;
```

```
    color: white;
```

```
}
```

```
/* date of match */
```

```
h2 {
```

```
    padding: 40px;
```

```
    font-size: 40px bold;
```

```
}
```

```
.match-detail {
```

```
    color: black;
```

```
    text-decoration: none;
```

```
}
```

```
.inrow {
```

```
    padding-top: 50px;
```

```
}
```

```
.menu a:hover {
```

```
    background-color: #200835;
```

```
    text-decoration: none;
```

```
}
```

```
.nav-link {
```

```
width: 300px;
margin: 0;
}
```

```
.nav-pills .active .nav-link:not(.active) {
  background-color: #200835;
}
```

```
.navbar {
  margin-top: 2px;
  display: flex;
  align-items: center;
  justify-content: center;
  padding-bottom: 0px;
}
```

```
.match-detail {
  align-content: center;
}
```

```
.container {
  width: 90vw;
  margin: 0 auto;
}
```

```
.container table {
  display: flex;

  justify-content: center;
}
```

```
.background {
  width: 100%;
  height: fit-content;
  background-color: #3d195b;
```

```

}
.image {
  display: flex;
  align-items: center;
  font-family: Regular, sans-serif, Georgia;
}
.image-position {
  background-color: white;
  padding: 10px 10px;
  width: 100px;
  height: 100px;
  background: white;
  -moz-border-radius: 70px;
  -webkit-border-radius: 70px;
  border-radius: 70px;
  margin-top: 10px;
}

h1 {
  margin-left: 30px;
  color: aliceblue;
  margin-top: 15px;
}
</style>

```

**ไฟล์ premeir-standing.vue**

```

<template>
  <div>
    <!-- title -->
    <div class="background-titlebar">

      <!-- title img logo and Premeir league text-->
      <div class="container">

```

```
<div class="image">
  <!-- ทำการเพิ่ม logo ของ League โดยใช้ <img> -->
  
  <h1>Premeir League</h1>
</div>
</div>
<!-- end title img logo and Premeir league text-->

<!-- nav menu bar -->
<div class="navbar">
  <!-- ทำเป็นเมนูสำหรับคลิกเลือก page โดยใช้ navbar -->
  <ul class="nav nav-pills nav-fill red">
    <li class="nav-item">
      <a class="nav-link " aria-current="page" href="/">Home</a>
    </li>
    <!-- เมื่อเราอยู่ตำแหน่งหน้าไหนจะให้ปุ่ม active ที่เมนูนั้นๆ -->
    <li class="nav-item active">
      <a class="nav-link" href="premeir-standing">Standing</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="premeir-schedule">Schedule</a>
    </li>
  </ul>
</div>
<!-- end of nav menu bar -->

</div>
<!-- end of title -->
```



```

<!-- body & table -->
<div class="container">

<!-- table -->
<div class="table-data borderless">

<!-- head table & columns -->
<!-- สร้างหัวตารางตามที่ได้ออกแบบไว้ -->
<table class="table ">
  <thead class="">
    <tr>
      <th scope="col">Pos</th>
      <th scope="col" colspan="2">Club</th>
      <th scope="col">Played</th>
      <th scope="col">Win</th>
      <th scope="col">Drawn</th>
      <th scope="col">Lose</th>
      <th scope="col">G.Score</th>
      <th scope="col">G.Against</th>
      <th scope="col">G.Different</th>
      <th scope="col">Pts</th>
    </tr>
  </thead>
  <!-- end head table & columns -->

  <!-- body data standing -->
  <tbody>
    <!-- วงวน loop เพื่อดึงค่าออกจากตัวแปร object ที่ได้ get ค่าจาก api มา -->
    <tr v-for="item in info.data.data" :key="item">
      <!-- นำข้อมูลที่ได้มาแสดงโดยใช้ตัวแปรที่ได้สร้างขึ้นมาเรียก object -->
      <th scope="row" class="pos-position">{{ item.position }}</th>
      <td class="team-image">
        
</td>
<td width="300px" class="team-name">{{ item.team_name }}</td>
<td width="180px" class="">{{ item.overall.games_played }}</td>
<td width="180px">{{ item.overall.won }}</td>
<td width="180px">{{ item.overall.draw }}</td>
<td width="180px">{{ item.overall.lost }}</td>
<td width="180px">{{ item.overall.goals_scored }}</td>
<td width="180px">{{ item.overall.goals_against }}</td>
<td width="180px">{{ item.overall.goals_diff }}</td>
<td width="100">{{ item.points }}</td>
</tr>
</tbody>
<!-- end body data standing -->

</table>
</div>
<!-- end table -->

</div>
<!-- end body & table -->
</div>
</template>

<script>
import axios from 'axios'
export default {
  data () {
    return {

```

```

    info: {}

    // JSON.stringify(jsArray) converts the jsArray into a string which can be stored in
    sessionStorage

    },
    // get api function
    // ใช้ axios เพื่อ get ค่าจาก api ของ backend
    mounted () {
        axios.get('https://soccerscoreapi.herokuapp.com/standings/352').then(response => {
            // สร้างตัวแปร object มาเพื่อเก็บค่า api
            this.info = response
        })
    }
}
</script>

```

```

<style scoped>
.table {
    font-size: 20px;
    font-family: "B612", sans-serif;
}

```

```

/* make table borderless */
.borderless th {
    border: none;
}
a {
    font-size: 25px;
    color: white;
}

```

```

/* hover of nav bar color */

```

```
a:hover {  
  background-color: #200835;  
}
```

```
/* make text in row go center */  
td , th{  
  text-align: center;  
  vertical-align: middle;  
}
```

```
/* size of nav menu */  
.nav-link {  
  width: 300px;  
  margin: 0;  
}
```

```
/* color of active nav menu */  
.nav-pills .active .nav-link:not(.active) {  
  background-color: #200835;  
}
```

```
/* set position of nav menu */  
.navbar {  
  margin-top: 2px;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  padding-bottom: 0px;  
}
```

```
/* set 1st columne to center */  
.pos-position {  
  padding-top: 22px;
```

```
}
```

```
/* set 2nd to center */
```

```
.team-image {  
  display: flex;  
  align-items: left;  
  justify-content: left;  
  padding-top: 50%;  
  padding-bottom: 50%;  
}
```

```
.team-name {  
  padding-top: 20px;  
  padding-left: 20px;  
  text-align: left;  
}
```

```
.team-logo {  
  width: 30px;  
  height: 30px;  
}
```

```
.tab-bar {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  color: black;  
  padding-left: 10px;  
  padding-right: 10px;  
  margin-top: 10px;  
}
```

```
.container {  
  width: 90vw;  
  margin: 0 auto;  
}
```

```
.background-titlebar {  
  width: 100%;  
  height: fit-content;  
  background-color: #3d195b;  
}  
  
.image {  
  display: flex;  
  align-items: center;  
  font-family: Regular, sans-serif, Georgia;  
}  
  
.image-position {  
  background-color: white;  
  padding: 10px 10px;  
  width: 100px;  
  height: 100px;  
  background: white;  
  -moz-border-radius: 70px;  
  -webkit-border-radius: 70px;  
  border-radius: 70px;  
  margin-top: 10px;  
}  
  
h1 {  
  margin-left: 30px;  
  color: aliceblue;  
  margin-top: 15px;  
}  
</style>
```

## ไฟล์ SoccerStadium.vue

```
<template>

  <!-- title and background -->
  <div class="vuetify">
    <!-- สร้าง Text กราฟฟิกโดยใช้ CSS -->
    <div class="plate">
      <p class="text">SOCCERSCORE</p>
    </div>
    <!-- end title and background -->

    <!-- img menu -->
    <div class="ct-center">
      <div class="row">
        <!-- ทำการลิง button โดยใช้ <a> -->
        <a href="premeir-standing" class="btn space">
          <!-- เพิ่มรูปภาพโดยใช้ <img> -->
          
        </a>
      </div>
      <div class="row">
        <!-- ทำการลิง button โดยใช้ <a> -->
        <a href="laliga-standing" class="btn">
          <!-- เพิ่มรูปภาพโดยใช้ <img> -->
          
        </a>
      </div>
    </div>
  </div>

  <!-- end img menu -->

</div>
```

```
</template>
```

```
<script>
```

```
export default {
```

```
  data () {
```

```
    return {}
```

```
  }
```

```
}
```

```
</script>
```

```
<style scoped>
```

```
#home {
```

```
  align-content: center;
```

```
  size: 60;
```

```
}
```

```
body {
```

```
  background-color: #c4c4c4;
```

```
}
```

```
/* grahfic title */
```

```
.text {
```

```
  position: absolute;
```

```
  top: 10%;
```

```
  right: 50%;
```

```
  transform: translate(50%, -50%);
```

```
  text-transform: uppercase;
```

```
  font-family: verdana;
```

```
  font-size: 50px;
```

```
  font-weight: 700;
```

```
  color: #f5f5f5;
```

```
  text-shadow: 1px 1px 1px #919191, 1px 2px 1px #919191, 1px 3px 1px #919191,
```

```
    1px 4px 1px #919191, 1px 5px 1px #919191, 1px 6px 1px #919191,
```

```
    1px 7px 1px #919191, 1px 8px 1px #919191, 1px 9px 1px #919191,
```



```
1px 10px 1px #919191, 1px 18px 6px rgba(16, 16, 16, 0.4),  
1px 22px 10px rgba(16, 16, 16, 0.2), 1px 25px 35px rgba(16, 16, 16, 0.2),  
1px 30px 60px rgba(16, 16, 16, 0.4);  
}
```

```
/* set img menu to center */
```

```
.ct-center {  
  display: flex;  
  justify-content: center;  
  flex-direction: column;  
  align-items: center;  
}
```

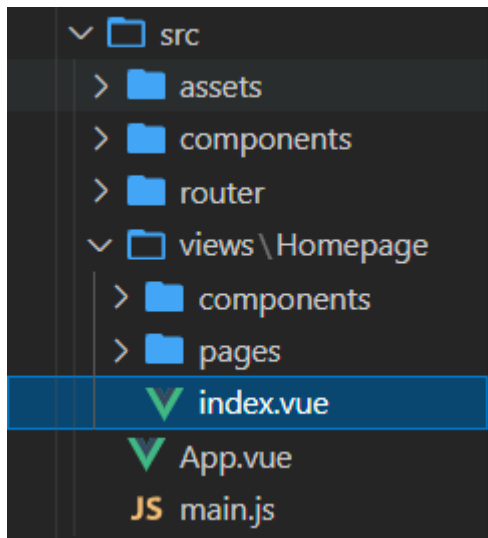
```
.space {  
  margin-top: 50px;  
  margin-bottom: 50px;  
}
```

```
.image-home-premeir {  
  width: 500px;  
  height: 200px;  
  opacity: 1;  
  border: 2px solid transparent;  
  border-radius: 30px;  
  background-color: aliceblue;  
  padding: 20px 50px;  
  margin-top: 140px;  
}
```

```
.image-home-laliga {  
  width: 500px;  
  height: 230px;  
  opacity: 1;  
  border: 2px solid transparent;  
  border-radius: 30px;
```

```
background-color: aliceblue;
background-position: center;
background-size: cover;
padding: 20px 50px;

}
/* background position */
.vuetify {
background-image: url(../../assets/background-homepage.jpg);
background-size: cover;
height: 100vh;
background-position: center;
}
</style>
```



มีอีก 1 ไฟล์ที่อยู่ใน Homepage คือ index.vue ใช้สำหรับการ router ในส่วนของ router

#### ไฟล์ index.vue

```
<template>
  <section id="Homepage">
    <router-view />
    <head>
      <H1>SOCCERSCORE</H1>
    </head>
  </section>
</template>
```

App.vue ก็เป็นส่วน router view เช่นกัน และส่วนสุดท้ายก็คือ main.js เป็นส่วนการ render view

#### ไฟล์ App.vue

```
<template>
  <section id="Homepage">
    <router-view />
    <head>
      <H1>SOCCERSCORE</H1>
    </head>
  </section>
</template>
```

## ข้อมูลของ API

ข้อมูลที่เก็บในฐานข้อมูลดึงมาจาก <https://app.sportdataapi.com/> โดยจะดึงมาทุกๆ 1 ชั่วโมง เพื่ออัปเดตฐานข้อมูล โดยในส่วนของ api ฟรีนั้นสามารถดึงข้อมูลได้ 2 ลีค และดึงได้เพียง 500 request ต่อสัปดาห์ ซึ่งสามารถเพิ่มจำนวนของ request และจำนวนของลีคได้ ขึ้นอยู่กับ Subscribe ที่เลือกไว้

API Key

Rotating your key will reset your API usage statistics.

Copy to Clipboard

Rotate Key

Used Requests

39 %

194 used / 500 available