**NodeJs Final Project Group 3 Final Tasks**

* Create Table for **players**
  + **ID**
  + **name -** Player full name
  + **DOB** - - Will be used to determine the age
  + **position -**  The wing of the player example central back (CB)
  + **club\_id -** Club the player plays for or last played for)
  + **nationality** - County of the person
  + **county\_id** - if player is liberian, what county. Can be null
  + **status** - if the player is active or retired
  + **market\_value** Player current values
  + **agent** - The Player Agent name
  + **photo** Photo of the user
* db.run("CREATE TABLE IF NOT EXISTS players (id INTEGER PRIMARY KEY AUTOINCREMENT,  fullname VARCHAR(50) NOT NULL,DOB DATE NOT NULL, position CHAR(10) NOT NULL, club\_id INTEGER NOT NULL, nationality INTEGER NOT NULL, county\_id INTEGER NULL, status VARCHAR(50) NOT NULL, market\_value INTEGER NULL, agent VARCHAR(50) NULL, photo VARCHAR(50) NULL)");
* Create a JSON file of the countries in the word and save as **countries.json** and **save** it in the data folder on the project
* Create a JSON file of all the counties in Liberia call **counties.json** and save it in the **data** folder on the project
* Create a table call **competitions** in the database
  + **id**
  + **competition** - name of the league
  + **country** - the county the league is located
  + **players** - number of players in the league (autogenerated)
  + **market**\_**value** - total market value of the league
  + **continent** - the continent the league is located
  + **logo** - the logo of the league
  + **founded** - year the league started
  + **type** - football, kickball, basket
* db.run("CREATE TABLE IF NOT EXISTS competitions (id INTEGER PRIMARY KEY AUTOINCREMENT,  competition VARCHAR(50) NOT NULL, country\_id INTEGER NOT NULL, players INTEGER NULL, market\_value INTEGER NULL, continent VARCHAR(50) NOT NULL, logo VARCHAR(50) NULL, founded DATE NULL, type VARCHAR(50) NOT NULL)");
* 1 Create table called clubs
  + **Id**
  + **club** - the name of the team
  + **country**\_**id** - the club country
  + **squad** - the number of team members
  + **stadium** - the stadium name
  + **market\_value** -the market value
  + **badge**
* db.run("CREATE TABLE IF NOT EXISTS clubs (id INTEGER PRIMARY KEY AUTOINCREMENT, club VARCHAR(50) NOT NULL, country\_id INTEGER NOT NULL, squad INTEGER NULL, stadium VARCHAR(50) NULL, market\_value INTEGER NULL, badge VARCHAR (50) NOT NULL )");
* create a table called **games** 
  + id
  + **home**  - the home team
  + **away**  - the away team
  + **start** - the start time of the game
  + **status** - if 1 the game is on, 2 it over, and 3 it postpones
  + **period** - if 1 it first half, 2 it half time and 3 it second half
  + **home\_goal** - home team goals, will update once someone scores
  + **away\_goal** - away team goals
  + **season\_id** - the season of the game
* db.run("CREATE TABLE IF NOT EXISTS games (id INTEGER PRIMARY KEY AUTOINCREMENT, home INTEGER NOT NULL, away INTEGER NOT NULL, start DATE NOT NULL, status INTEGER NOT NULL, period INTEGER NOT NULL, home\_goal INTEGER NOT NULL, away\_goal INTEGER NOT NULL, season\_id INTEGER NOT NULL)");
* create a table call **cards**
  + id
  + **player\_id** - id of player booked
  + **game\_id** - id of the game booked in
  + **card\_type** -the type of the card, yellow or red
  + **minutes** - time in the game player was booked
* db.run("CREATE TABLE IF NOT EXISTS cards (id INTEGER PRIMARY KEY AUTOINCREMENT, player\_id INTEGER NOT NULL, game\_id INTEGER NOT NULL, type VARCHAR(50) NOT NULL, minutes CHAR(10) NOT NULL)");
* create a table call **activities**
  + id
  + **game\_id** - the game of the corner
  + **team\_id** - the that won the corner
  + **type** - the type of the activity
  + **minutes** - the minutes in the game the corner happened
* db.run("CREATE TABLE IF NOT EXISTS activities (id INTEGER PRIMARY KEY AUTOINCREMENT, game\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, type VARCHAR(50) NOT NULL, minutes CHAR(10) NOT NULL)");

this can be foul, corner,possession,throws,penalty

* create a table call **substitutions**
  + id
  + **game\_id** - the game
  + **off** - the player off
  + **in** - the player in
  + **team\_id** - the team id
  + **minutes** - the minutes of the sub
* db.run("CREATE TABLE IF NOT EXISTS substitutions (id INTEGER PRIMARY KEY AUTOINCREMENT, game\_id INTEGER NOT NULL, player\_off INTEGER NOT NULL, player\_in INTEGER NOT NULL , team\_id INTEGER NOT NULL, minutes VARCHAR(50) NOT NULL)");
* create a pages for the **search.ejs** when the user type in the search bar that is on the top nav let the result of the search display on the search page. Implement querying the database searched terms. tables to search are, players, clubs, games, completions. Use the table layout on <https://www.transfermarkt.com/schnellsuche/ergebnis/schnellsuche?query=chelsea> to display the results on the page. Let the results or the query be sent to url like /search?q=watanga
* create a table call **seasons**
  + id
  + **start** - the year the league start
  + **end** - the time the league end
  + **team** - the number of teams in the search
  + **status** – false is finished, true is on
  + **games** - number of games played
* db.run("CREATE TABLE IF NOT EXISTS seasons (id INTEGER PRIMARY KEY AUTOINCREMENT, start DATE NOT NULL, end DATE NOT NULL, teams INTEGER NOT NULL , status BOOLEAN NOT NULL, games INTEGER NOT NULL)");
* create a table call **phase**
  + id
  + **season\_id** – the season of the edition
  + **team\_id** the name of the team in the edition
  + **status** if season ends, edition ends
* db.run("CREATE TABLE IF NOT EXISTS phases (id INTEGER PRIMARY KEY AUTOINCREMENT, season\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, status BOOLEAN NOT NULL)");
* create a table call **standings**
  + id
  + **season\_id** - the id of the season
  + **team\_id** - the team in the season
  + **play** - the number of games played
  + **win** - the number of game won
  + **loss** - games loss
  + **draw** - games drew
  + **goals\_for** - goals score
  + **goals\_against** -
  + **pts** - total points accumulated
* db.run("CREATE TABLE IF NOT EXISTS standings (id INTEGER PRIMARY KEY AUTOINCREMENT, season\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, play INTEGER NULL, win INTEGER NULL, loss INTEGER NULL, play draw INTEGER NULL, goal\_for INTEGER NOT NULL, goal\_against INTEGER NULL, points INTEGER NULL)");
* create a table call **scorers**
  + id
  + **player\_id** the player who scores
  + **game\_id** the game the player scored
  + **goals** the number of goals scored
  + **minutes**
* db.run("CREATE TABLE IF NOT EXISTS scorers (id INTEGER PRIMARY KEY AUTOINCREMENT, player\_id INTEGER NOT NULL, game\_id INTEGER NOT NULL, goal INTEGER NOT NULL, minutes VARCHAR(50) NULL)");
* create table call **champions**
  + id
  + **competition\_id -** id of the completions
  + **season\_id -** id season
  + **club\_id -** club that won
* db.run("CREATE TABLE IF NOT EXISTS champions (id INTEGER PRIMARY KEY AUTOINCREMENT, competition\_id INTEGER NOT NULL, season\_id INTEGER NOT NULL, club\_id INTEGER NOT NULL, trophy INTEGER NOT NULL)");
* 1 - create a table call countries
  + **id**
  + **country -** name of the country
  + **flag -** flag of the country – url
* db.run("CREATE TABLE IF NOT EXISTS countries (id INTEGER PRIMARY KEY AUTOINCREMENT, country VARCHAAR(50) NOT NULL, flag VARCHAR(50) NOT NULL)");
* create a table call lineup
  + **id**
  + **game\_id** - the game the line-up is for
  + **team\_id** - the lineup team
  + **Player\_id** - the player
  + **Position** - the wing/position of the player
  + **Start** - yes if the player starts or no if not
* db.run("CREATE TABLE IF NOT EXISTS lineups (id INTEGER PRIMARY KEY AUTOINCREMENT, game\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, player\_id INTEGER NOT NULL, number INTEGER NULL, position NULL, start BOOLEAN NOT NULL)");

**TOURNAMENT**

* create database for tournament **tournament.db**
* create a table call **tournaments**
  + ID - - will be unique id
  + Name - - will be the name of the tournament
  + Badge
  + Founded
  + Country
  + Continent
* Create a table call **tournaments**
  + Id
  + Name - name of the edition
  + badge
  + Host
  + Start - start date
  + End - the end date
  + Winner - the id of the winning team
  + Runner-up - id of the runner-up team
  + Third-place -id of the third-place team
* db.run("CREATE TABLE IF NOT EXISTS tournaments (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(50) NOT NULL, badge VARCHAR(50) NOT NULL, host VARCHAR(50) NOT NULL, start DATE NOT NULL, end DATE NOT NULL, winner INTEGER NULL, runner\_up INTEGER NULL, third\_place INTEGER NULL)");
* Create a table call **teams**
  + Id - will be the unique id
  + **Name** - the name of the team
  + **Badge** - the flag or logo of the team
* db.run("CREATE TABLE IF NOT EXISTS teams (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(50) NOT NULL, badge VARCHAR(50) NOT NULL)");
* Create a table call **players**
  + **Id** -
  + **name** - full name of the player
  + **position** - the position the player
  + **photo** - the place
* db.run("CREATE TABLE IF NOT EXISTS players (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL, photo VARCHAR(50) NOT NULL)");
* Create a table call **fixtures** 
  + **Id**
  + **Tournamentid**
  + **start**
  + **Home**
  + **Away**
  + **Home-goal**
  + **Away-goal**
* db.run("CREATE TABLE IF NOT EXISTS fixtures (id INTEGER PRIMARY KEY AUTOINCREMENT, tournament\_id INTEGER NOT NULL, start DATE NOT NULL, home\_team INTEGER NOT NULL, away\_team INTEGER NOT NULL, home\_goals INTEGER NULL, away\_goals INTEGER NULL)");
* Lineup
  + Id
  + Fixtureid
  + Teamid
  + Playerid
  + Started
* db.run("CREATE TABLE IF NOT EXISTS lineups (id INTEGER PRIMARY KEY AUTOINCREMENT, fixture\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, player\_id INTEGER NOT NULL, started BOOLEAN NULL)");
* **Group**
  + **Id**
  + **Teamid**
  + **Group**
* db.run("CREATE TABLE IF NOT EXISTS groups (id INTEGER PRIMARY KEY AUTOINCREMENT, team\_id INTEGER NOT NULL, group VARCHAR(50) NOT NULL )");
* **Stages**
  + **id**
  + tournamentid
  + Stage
  + Home-team
  + Away-team
  + Start
* db.run("CREATE TABLE IF NOT EXISTS stages (id INTEGER PRIMARY KEY AUTOINCREMENT, tournament\_id INTEGER NOT NULL, stage VARCHAR(50) NOT NULL, home\_team INTEGER NOT NULL, away\_team INTEGER NOT NULL, date DATE NOT NULL)");
* create a table call **scorers**
  + id
  + **player\_id** the player who scores
  + **fixtureid** the game the player scored
  + **goals** the number of goals scored
  + **minutes**
* db.run("CREATE TABLE IF NOT EXISTS scorers (id INTEGER PRIMARY KEY AUTOINCREMENT, player\_id INTEGER NOT NULL, fixture\_team INTEGER NOT NULL, goals INTEGER NOT NULL, minutes CHAR(10) NOT NULL)");
* ACTIVITIES
* db.run("CREATE TABLE IF NOT EXISTS activities (id INTEGER PRIMARY KEY AUTOINCREMENT, fixture\_id INTEGER NOT NULL, team\_id INTEGER NOT NULL, type VARCHAR(50) NOT NULL, minutes CHAR(10) NOT NULL)");
* substitution
* db.run("CREATE TABLE IF NOT EXISTS substitutions (id INTEGER PRIMARY KEY AUTOINCREMENT, fixture\_id INTEGER NOT NULL, player\_off INTEGER NOT NULL, player\_in INTEGER NOT NULL , team\_id INTEGER NOT NULL, minutes CHAR(10) NOT NULL)");
* extra time
* db.run("CREATE TABLE IF NOT EXISTS extra\_times (id INTEGER PRIMARY KEY AUTOINCREMENT, fixture\_id INTEGER NOT NULL, added\_time INTEGER NOT NULL, period CHAR(10) NOT NULL)");