



Al Imam Mohammad Ibn Saud Islamic University  
College of Computer and Information Sciences  
**Computer Science Department**

# SportX

**CS346 – Web Development Project**  
**Second semester 2022-2023**

STUDENT NAME	ID
Fatimah Ahmed Bin Dous	440026128
Nouf Alkernass	441019825
Noura Omar Alosaimi	440022233
Ahlam Alharbi	441020829

Supervised by: Amal Algefes  
Due 12/2/2023



SportX unlocks the world of private home courts and unique playing experiences to the general public, Players and coaches.

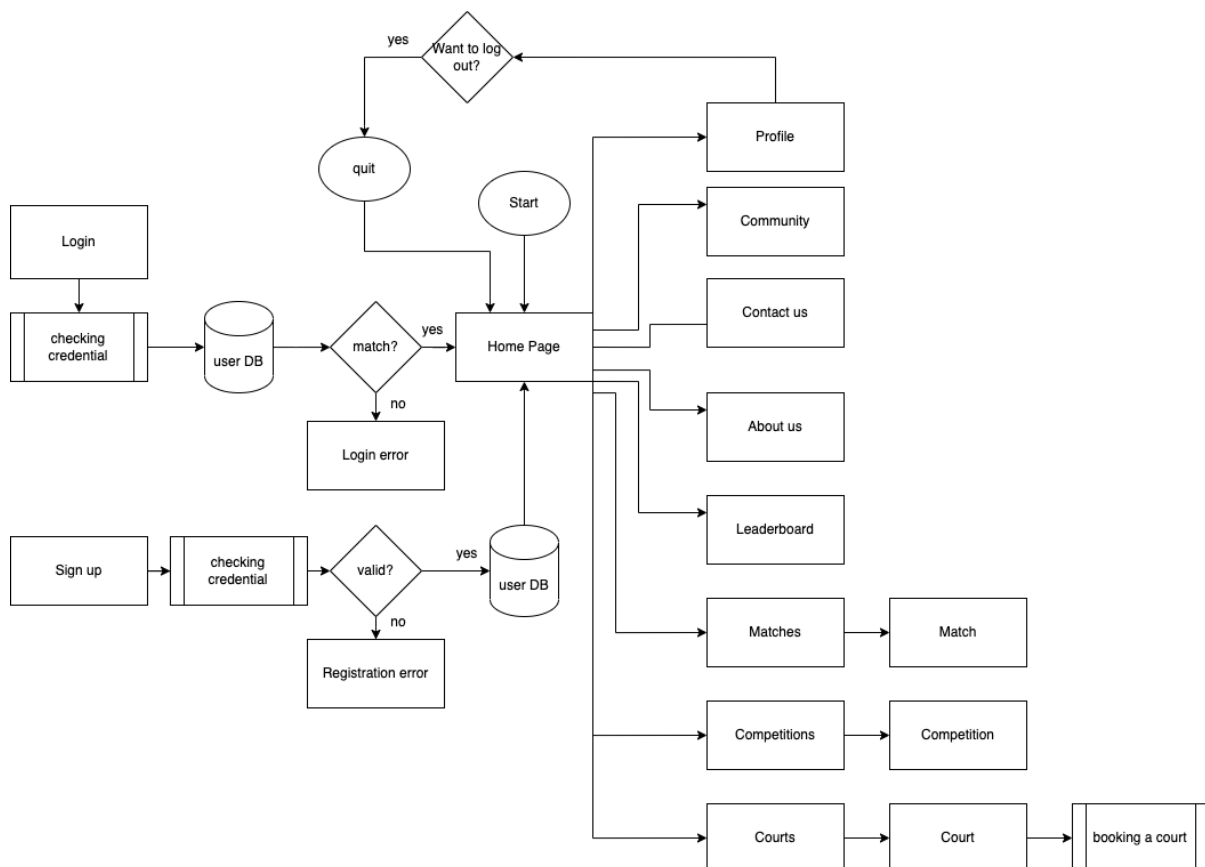
#### Research goals

- Increase the utilization of the venues.
- Introduce a pooling model to enable very competitive price.
- Try to attract an audience who would like to practice sports but don't have teams to play with.
- Use competitions and prize money to entice teams and individuals
- Apply gamification mechanics to ensure retention.

#### Problem statement

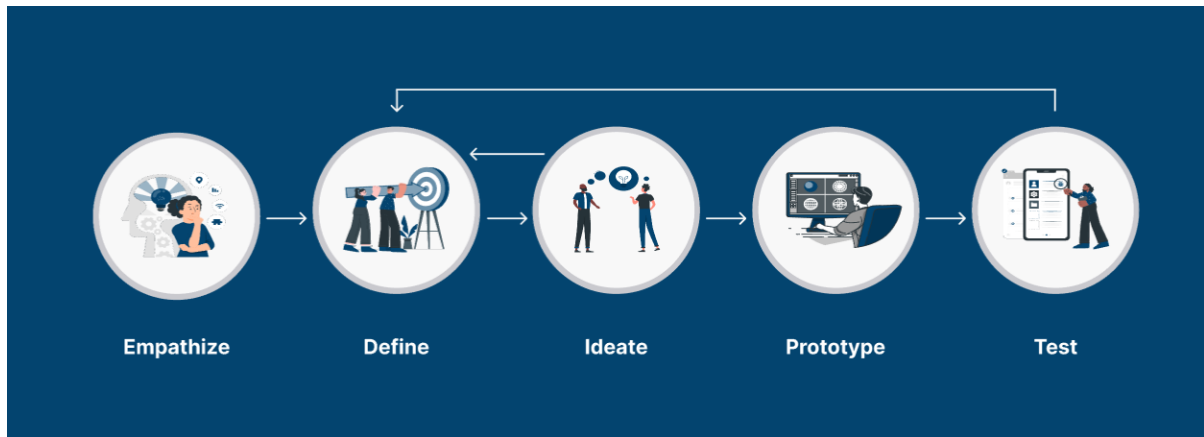
A club owner wants to increase his revenues by start renting venues and courts to the public who want to use them for different activities and he wants to organize this process through a mobile app. The app should enable users to book, pay, cancel and share (pooling) the rented venues with each other. The owner hopes in a later phase that it would be a platform that other venues owners would use to utilize their venues just like him.

## 1. Flow Chart

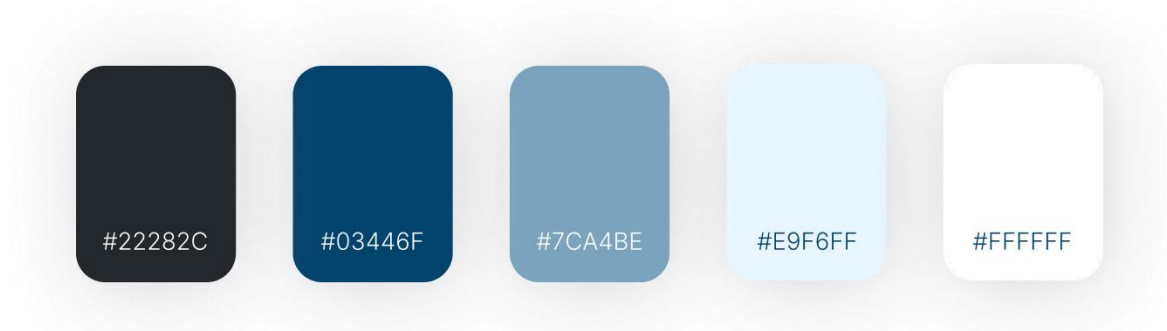


## 2. Look & Feel

We used the design thinking process, questioning assumptions, developing prototypes, and testing them iteratively to understand users, challenge assumptions, and redefine problems.



### Color Palette



### Typography

Poppins	H1 heading	H1 heading	H1 heading	H1 heading
	Inter / Semi Bold / 24px	Inter / Medium / 24px	Inter / Regular / 24px	Inter / Light / 24px
	H2 heading	H2 heading	H2 heading	H2 heading
	Inter / Semi Bold / 18px	Inter / Medium / 18px	Inter / Regular / 18px	Inter / Light / 18px
	H3 heading	H3 heading	H3 heading	H3 heading
	Inter / Semi Bold / 14px	Inter / Medium / 14px	Inter / Regular / 14px	Inter / Light / 14px
	H4 heading	H4 heading	H4 heading	H4 heading
	Inter / Semi Bold / 12px	Inter / Medium / 12px	Inter / Regular / 12px	Inter / Light / 12px



[cite any resources and references used to create the HTML and CSS style sheets]

[https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)

<https://www.geeksforgeeks.org/types-of-css-cascading-style-sheet/>

<https://blog.stackfindover.com/category/html-css/>

### 3. Dynamic Components

- basketball.html
- Football.html
- Swimming.html
- tennis.html
- Profile.html
- LeaderBoard.html

### 4. Business Logic

The three data blocks that comprise the core of our database are a user's name, an email address, and a password.

We just created a simple nodejs (express) server and connected that server to a MongoDB, then at signup page we simply put all data (name , email and password) from user and save that data to MongoDB then at login we get email and password from user (login page) and match that email and password with the stored email in mongoddb.

```
//route for homepage
app.get("/", checkUser, async(req, res) =>{
  console.log(req.cookies)
  res.render('Homepage' , {user: null})
})
```

The app.get method routes the HTTP GET Requests to the path which is being specified with the specified callback functions.

```
// here nodejs server will recive name email and password
app.post("/register" , async(req, res) =>{
  const {name , email , password} = req.body
  console.log(req.body)
  const hashPassword = await bcrypt.hash(password , 10)
  const data = new userModel({
    name,
    email ,
    password : hashPassword
  })
})
```

we use the app.post method and then we specify the route, and we have a callback.



```
_id: ObjectId('63e7f81ab17bb607fec5cbe5')
name: "Fatimah Ahmed"
email: "faadous@outlook.com"
password: "$2b$10$/8E0IZG7U.4puczxxu4.v.q/g0T.csMqkN50YQYruuDx5Q9hw08G"
__v: 0
```

```
_id: ObjectId('63e7faf5b17bb607fec5cbe9')
name: "nouf"
email: "nouf@gmail.com"
password: "$2b$10$/aLSc28XLbns3vQJj0TtZ0TdhzASoxQ6ztOKqMsNSUCAXKtJTJcq"
__v: 0
```

```
_id: ObjectId('63e7fbcfb17bb607fec5cbed')
name: "Ahlam "
email: "ahlam@gmail.jf"
password: "$2b$10$/aLSc28XLbns3vQJj0TtZ0TdhzASoxQ6ztOKqMsNSUCAXKtJTJcq"
__v: 0
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\96650\Downloads\final> node app.js
(node:35144) [MONGODB] DeprecationWarning: Mongoose: the `strictQuery` option is deprecated. Please use `mongoose.set('strictQuery', true);` to suppress this warning.
Change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
server is running on port 5000 , http://localhost:5000
atlass db connected
63e7f0f40d4da00cc173eeeb
[Object: null prototype] {
  name: 'Fatimah Ahmed',
  email: 'faadous@outlook.com',
  password: '123456'
}
[Object: null prototype] {
  name: 'nouf',
  email: 'nouf@gmail.com',
  password: '112233'
}
[Object: null prototype] {
  name: 'Ahlam ',
  email: 'ahlam@gmail.jf',
  password: '654321'
}
}
```

## References

<https://github.com/mudit023/templates>

<https://getcssscan.com/css-box-shadow-examples>

<https://youtu.be/ZfZrcP0ymcA>

<https://youtu.be/OtooBEW9VIA>

<https://youtu.be/R8uMe8v5vjg>

[https://youtu.be/4x\\_Hb308pX4](https://youtu.be/4x_Hb308pX4)

<https://youtu.be/igNdxBdrTn0>

<https://youtu.be/fbYExfeFsI0>

<https://www.youtube.com/watch?v=S4fi6Qux-4g>

<https://www.youtube.com/watch?v=s0anSjEeua8>

<https://dev.to/codeclown/styling-a-native-date-input-into-a-custom-no-library-datepicker-2in>

## The Presentation

[click on the image to open the link]



