

# **Project Title: Game ON(Increment-3)**

**Team Lead:** Golla Ranga Kousik

## **Team Members(Project Group-16):**

Golla Ranga Kousik - 16322459

Nagula Ajay Kumar - 16327661

Mamjala Manisha - 16307995

Chenna Srinivas - 16307984

## **Story of the project and its Details:**

This Application helps those Individuals who are interested in sports or fun activities like cycling but are unable to find playgrounds or players to engage. we developed a web application through which these group of people can book a playing venue or create an activity in their preferred location as per their interest so that the others like him can join him. Our primary agenda is to generate a playfield in which you can fit in with other fellow misfits.

## **Significance:**

- This application is significant because it allows you to play when you want, where you want and how you want without any obstacles.
- Through our application one can rent their resources to others who have the group players to begin the game but not the things required to start their game
- It allows us to book any sports venue at a preferred location

## **Solution Approach:**

### **Waterfall Model**

- Requirements
- Design
- Implementation
- Verification
- Maintenance

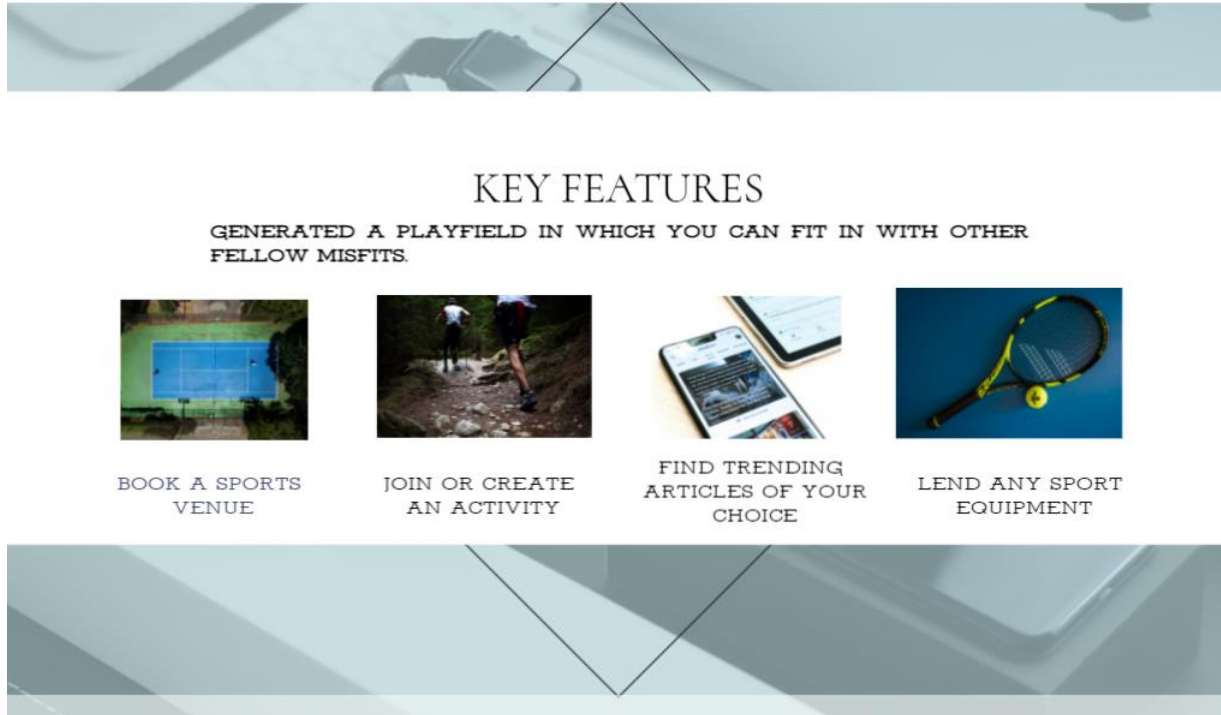
**Tech stack:** HTML5, CSS3, JavaScript, express, mangoDB

**Server:** node JS, JSON-Server, Live-server

**Others:** Local Storage, Google Map Embedded link

## Objectives:

- Develop a Web Application by embedding the below mentioned features and create a user-friendly interface with high scalability.

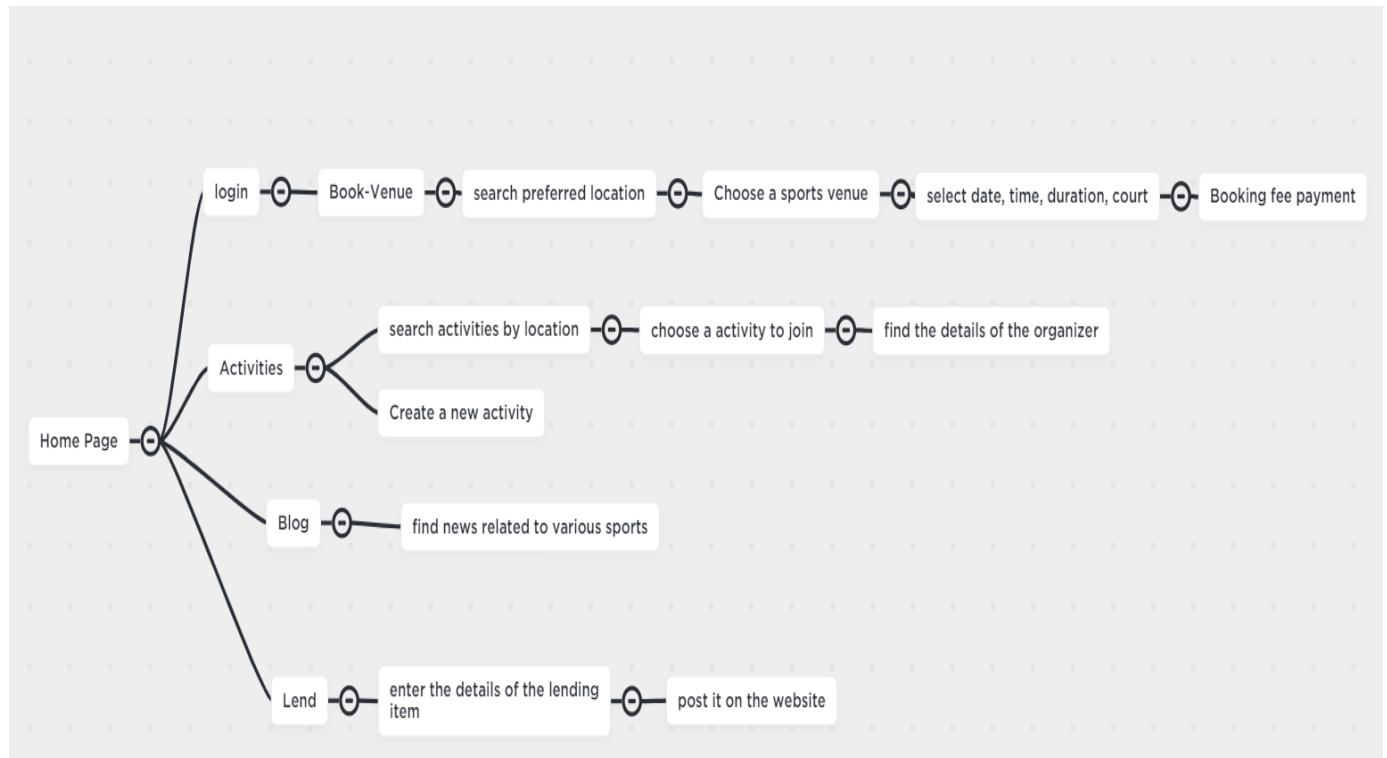


## Related Work:



- Meetup is a social networking site that permits you to find and join communities related to your own personal interests. However, rather than simply discussing these interests online, Meetup is utilized to arrange offline and in-person meetings in your area.
- This application allows us to create any outdoor or travel events at your preferred location but user cannot book a venue according to their preferred activity.
- Our application is mainly targeted towards the users who are interested in sports so we provided the booking, lending and activity features. users can also find the latest news about their favorite games through our blog page

## Flow of Website



### Previous Increment:

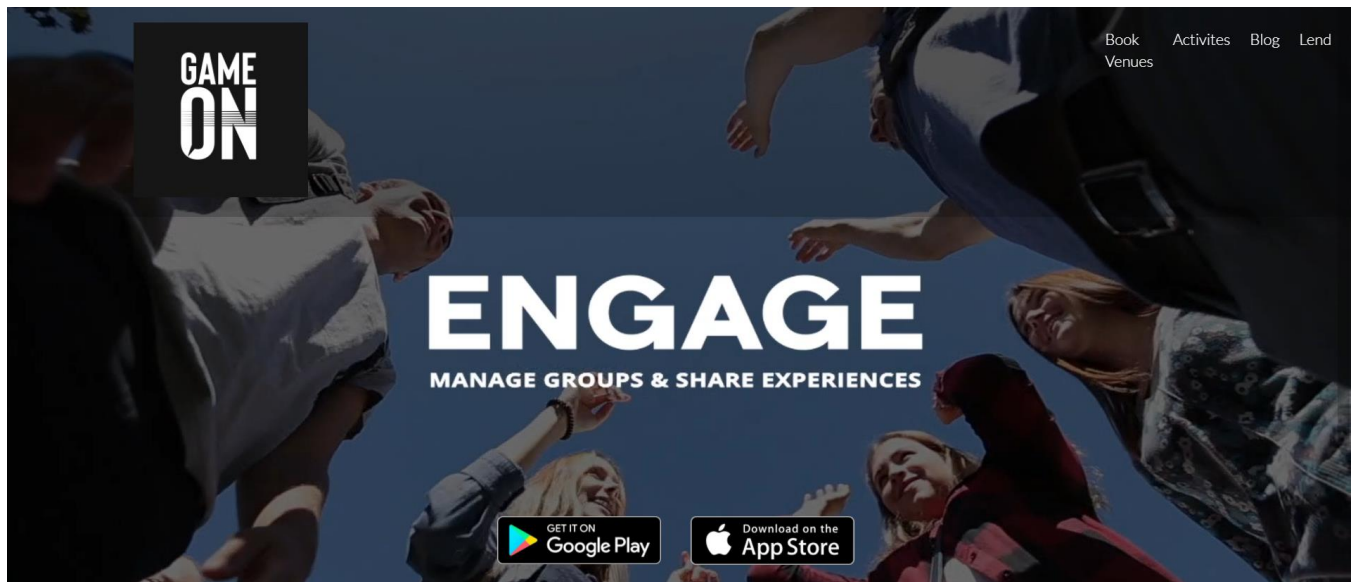
- Dependency Identification
- Performed risk analysis and developed a risk management strategy
- Assigned roles and responsibilities
- Prototype Development and creation of few wireframes
- Smoke testing of few major functionalities

### Current Increment:

- Complete development of left-over screens
- End-to-end feature testing of various functionalities
- Implementation of Debouncing search feature
- Real-time news fetching with API and search news feature
- Real-time random quote generator with bodybuilding quotes API (source- Rapid API)
- Log in on any page and ability to update all the user details and ability to view orders made by the user in the profile section.
- Acquire user feedback through the contact us page by utilizing form spree.
- Email feature with nodemailer.
- Filter functionality in the product page.

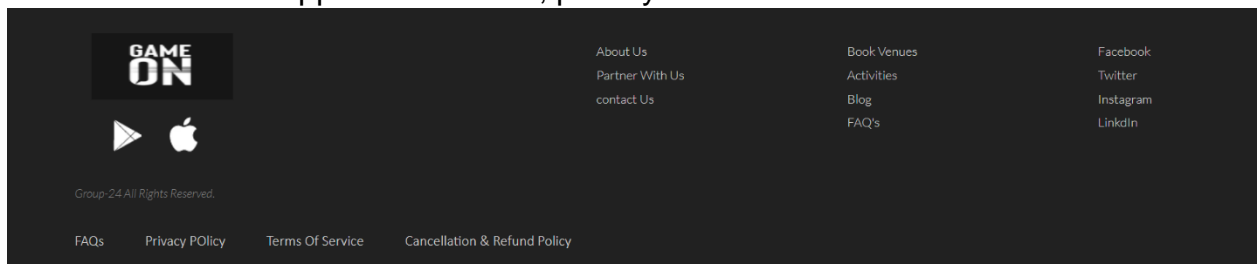
- Implemented MongoDB backend for login feature
- Admin dashboard with the ability to add data and view user details
- Integrated payment gateway using razor pay api

## Home Page:

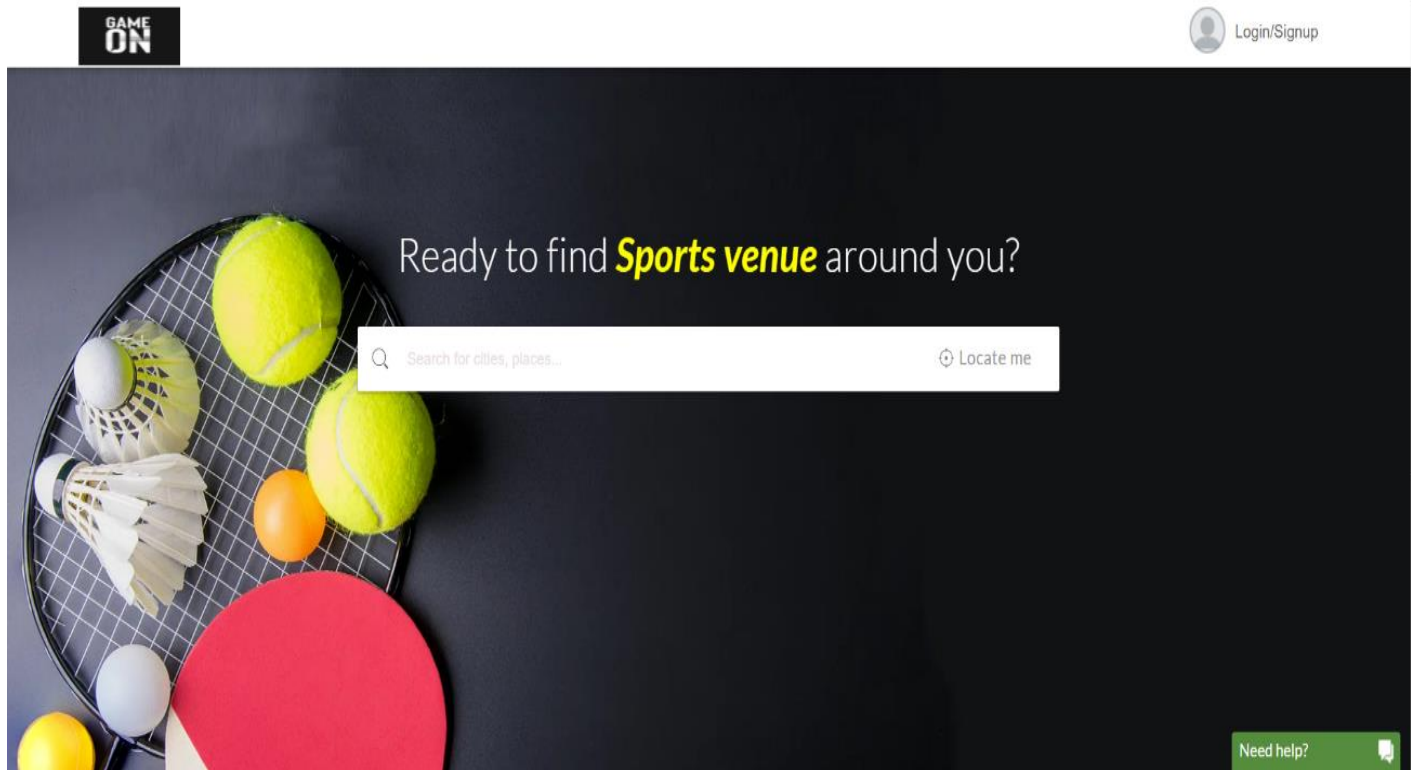


Contains website navigation menu and each element in the menu has link to its internal page

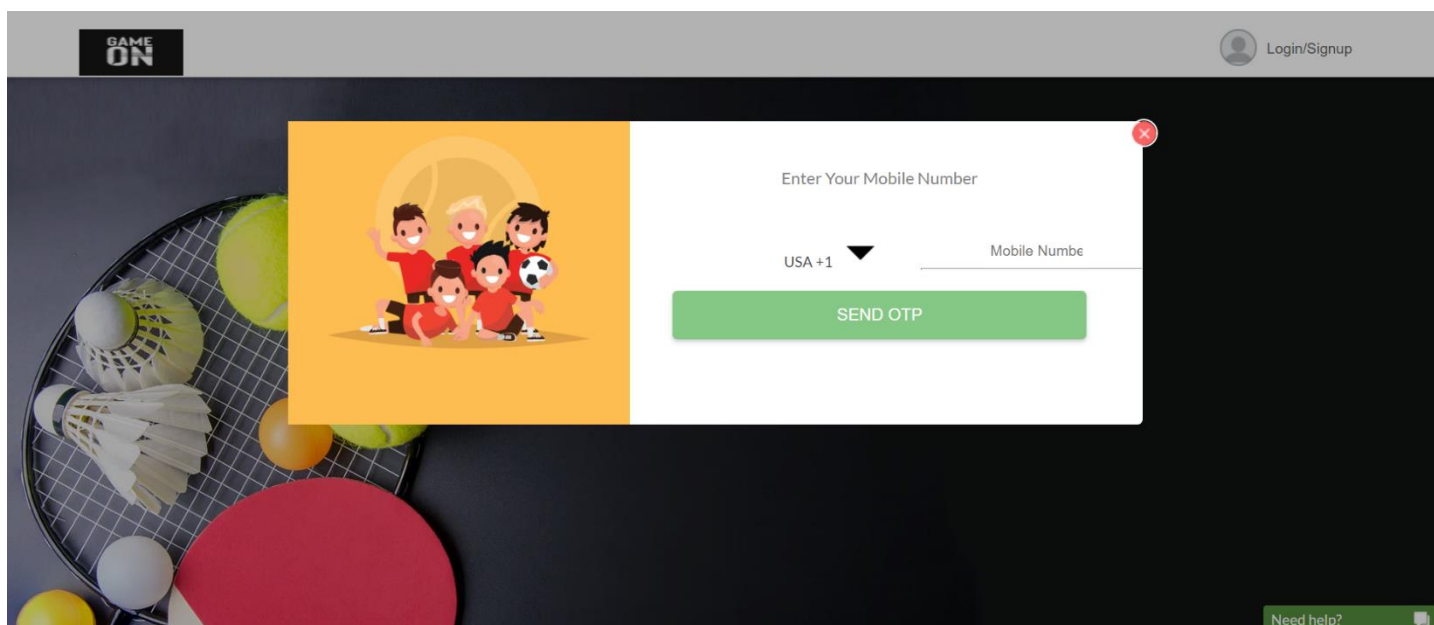
- Two optional responsive icons google play and app store are created to provide an opportunity for further development
- Every inner page contains the logo image that will redirect the user to the home page when clicked on it
- **Footer:** Essential support information, privacy and social media information.



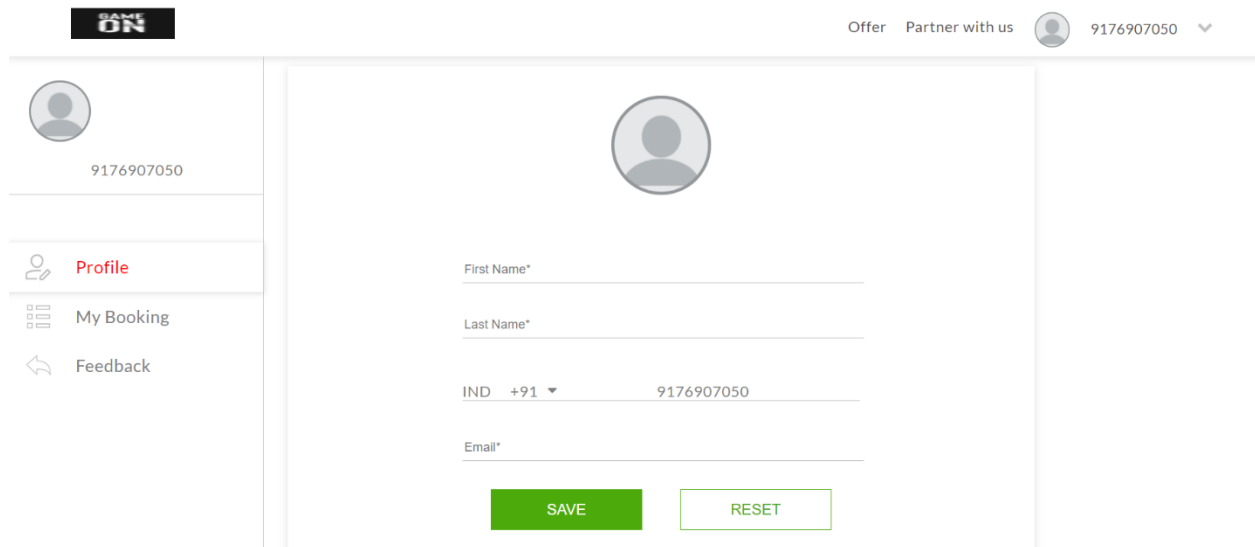
- **Book-Venues:** To find the nearby venues by passing the location information in search box.



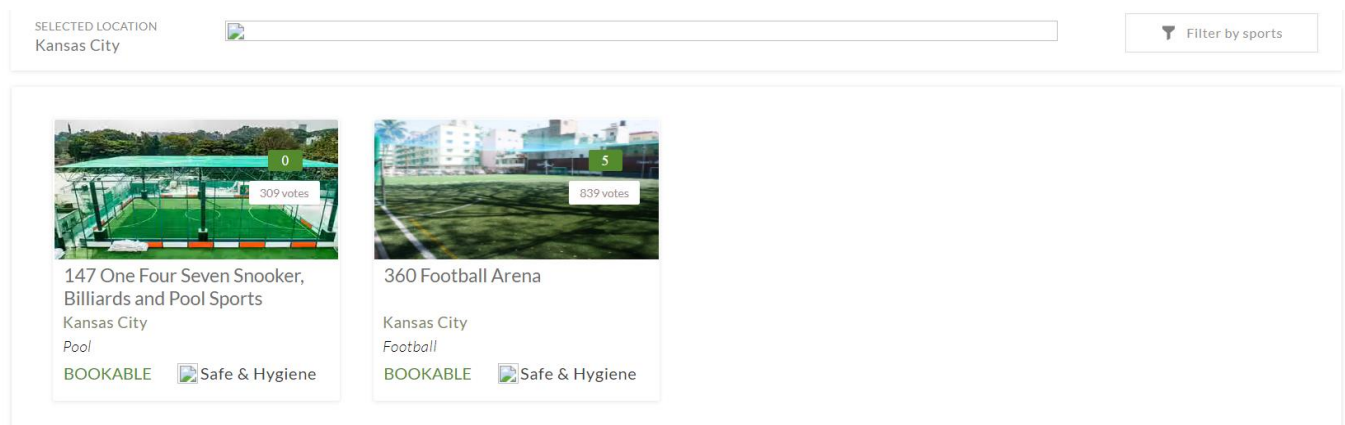
- Login popup window will be displayed when user clicks on the login/signup element
- User has to enter the valid mobile number and OTP/Password



- User entered otp will be validated and user will be directed to the login profile where he can enter personal information.
- Reset option is provided to erase all the information entered.
- In my booking section user can find the information related to his bookings



- Search results for Kansas City




- Filter feature for search results

SELECTED LOCATION

Kansas City


Filter by sports



0

236 votes

147 One Four Seven Snooker, Billiards and Pool Sports  
Kansas City  
Pool  
BOOKABLE Safe & Hygiene



5

327 votes

360 Football Arena  
Kansas City  
Football  
BOOKABLE Safe & Hygiene

Popular Sports

Badminton

Football

Cricket

Tennis

Table Tennis

Basketball

Running

Cycling

RESET

APPLY

## Details of the selected venue

Safe & Hygiene

Know More

Sanitized Court

Mask Are Must

Temperature Check

Handwash Station

SPORTS AVAILABLE

Pool

ADDITIONAL INFO

Football:

- It is recommended but not compulsory to wear football studs while playing at the facility.
- Metal studs are not allowed.

AMENITIES

Valet Parking

Restrooms

Dressing Room

Sitting Stand

Gym

First Aid

Shower

Drinking Water


TIMING

6 AM-12 AM

LOCATION

South United Football Club, RBANM's Grounds, Gate No 3, Gangadhar Chetty Road, near Ulsoor, Bengaluru, Karnataka

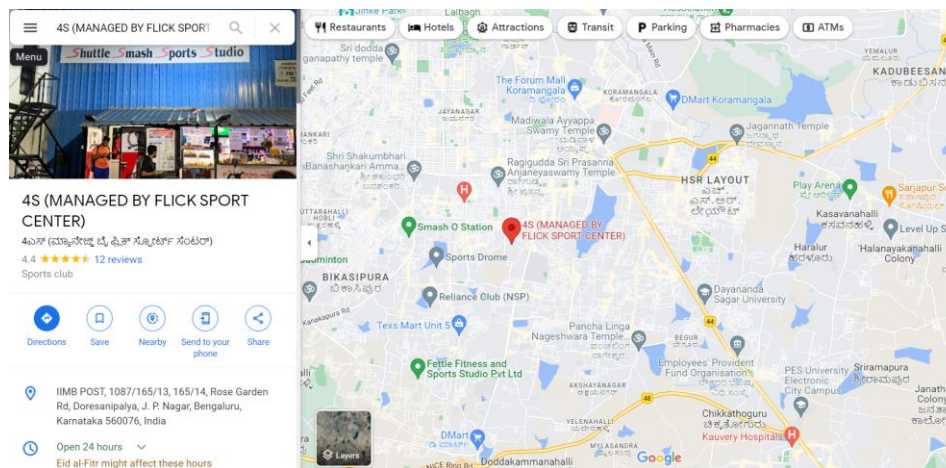
View larger map



Share

Book Now


Find the location of the venue by clicking on the google maps icon





- When clicked on the book now button, User will be redirected to the below page where he has to select date, sport, time, duration, court and add it in to the cart

SELECT SPORT



Pool

SELECT DATE

Today 22, Jan	<b>Tomorrow 23, Jan</b>	Monday 24, Jan	Tuesday 25, Jan	Wednesday 26, Jan	Thursday 27, Jan	Friday 28, Jan	Saturday 29, Jan
------------------	-----------------------------	-------------------	--------------------	----------------------	---------------------	-------------------	---------------------

SELECT START TIME

12.00 AM 05.00 AM	<b>05.00 AM 12.00 PM</b>	12.00 PM 05.00 PM	05.00 PM 12.00 AM					
05.00 PM	05.30 PM	06.00 PM	06.30 PM	07.00 PM	07.30 PM	08.00 PM	08.30 PM	09.00 PM
09.30 PM	10.00 PM	10.30 PM	11.00 PM	11.30 PM				

SELECT DURATION

-

1 hr

+

SELECT COURT

9 a Side Court 1	<b>9 a Side Court 2</b>	7 a Side Court 1	7 a Side Court 2	6 a Side Court 1
6 a Side Court 2	9 a Side Court 1			




Total cost for these booking - INR 2200


ADD TO CART

PROCEED

MY CART (1)

Football

 6 a Side Court 1
  22, January 2022
  06.00 PM to 07.00 PM

 INR 2200

INR 2200

CHECKOUT

127.0.0.1:8080 says  
please login

OK

4S Shuttle Smash Sports Studio - Managed by Flick Sport Center

SELECT SPORT



Badminton

SELECT DATE

Today 22, Jan	<b>Tomorrow 23, Jan</b>	Monday 24, Jan	Tuesday 25, Jan	Wednesday 26, Jan	Thursday 27, Jan	Friday 28, Jan	Saturday 29, Jan
------------------	-----------------------------	-------------------	--------------------	----------------------	---------------------	-------------------	---------------------

SELECT START TIME

12.00 AM 05.00 AM	<b>05.00 AM 12.00 PM</b>	12.00 PM 05.00 PM	05.00 PM 12.00 AM					
05.00 PM	05.30 PM	06.00 PM	06.30 PM	07.00 PM	07.30 PM	08.00 PM	08.30 PM	09.00 PM
09.30 PM	10.00 PM	10.30 PM	11.00 PM	11.30 PM				

SELECT DURATION

-

1 hr

+

MY CART (1)

Football

 6 a Side Court 1
  22, January 2022
  06.00 PM to 07.00 PM

 INR 2200

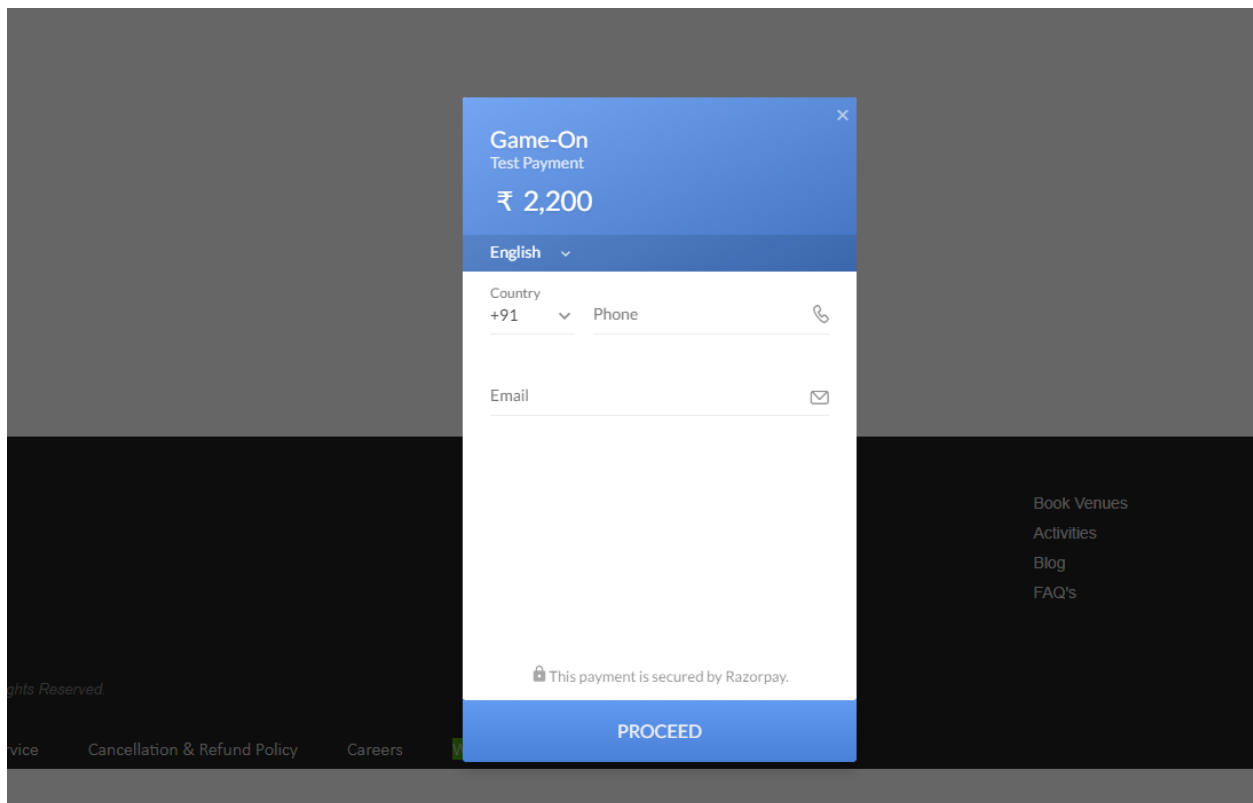
INR 2200

CHECKOUT

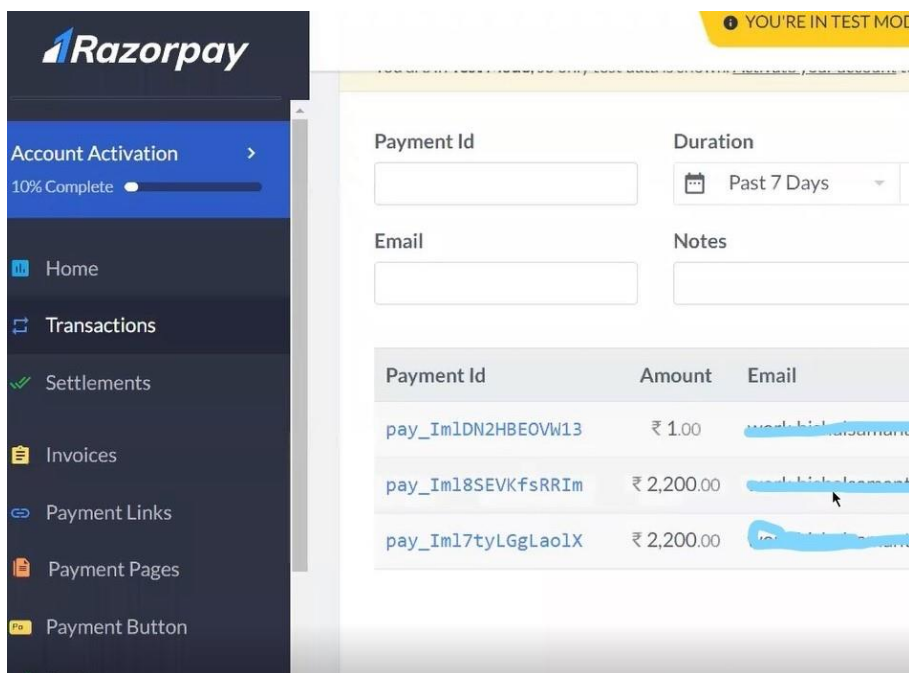
- An alert will be displayed when a user tries to complete the payment without logging in.



- Payment gateway will open when user clicks on the checkout



- Payment will reflect in your razor pay account after a successful payment



## Activities page:

- Activities in the region will be displayed when user enter any city name and clicks on the detect button

**GAME ON**

All cities

Kansas city

New york

Chicago


Florida

Las Vegas


- Activity results for Kansas City

Location  
**Kansas City**

Search  
For example, "Football"

 Sport  
**Select Sports** ▼


Categories  
**All** ▼



**Coaching**

**Daniel Cricket Academy**  
Kansas City  
6 AM - 8 AM, 4 PM - 8 PM

daniel cricket academy coaching



**Coaching**

**ST. JOHNS CRICKET ACADEMY**  
Kansas City  
Morning & Evening sessions

st. johns cricket academy coaching

- Click on any activity to find the contact details of the organizer



## Daniel Cricket Academy

6 AM - 8 AM, 4 PM - 8 PM

Kansas City

CALL

### Overview

DANIEL KANAPARTHI, a Level II certified Coach from England and Wales Cricket Board (ECB, United Kingdom), is always passionate for his all time favourite dream game "CRICKET", has opted it as a career and opened his own Academy - DANIEL'S CRICKET ACADEMY in

- Creating activity at a preferred location

### Create Activity

Location

Dallas

Football

Event Date

04/20/2022

12:20

Submit

Create



#### Major Fun

Location: Kansas City  
Event Date: 2022-04-07  
Event Time: 11:30 PM



#### Football

Location: Dallas  
Event Date: 2022-04-12  
Event Time: 12:20



#### Soccar

Location: Overland Park  
Event Date: 2022-04-12  
Event Time: 12:42



#### Soccar

Location: Overland Park  
Event Date: 2022-04-12  
Event Time: 12:22

- Lending feature to lend any equipment

### Lend Item

Location  
Overland Park

Sport Item  
Wickets and bases


33

Start Date  
04/11/2022


End Date  
04/14/2022

Submit


Create




**Balls**  
Price: \$55/day  
Location: Dallas  
Event Start Date: 2022-04-28  
Event End Date: 2022-04-30





**Wickets and bases**  
Price: \$88/day  
Location: Kansas City  
Event Start Date: 2022-04-14  
Event End Date: 2022-04-23



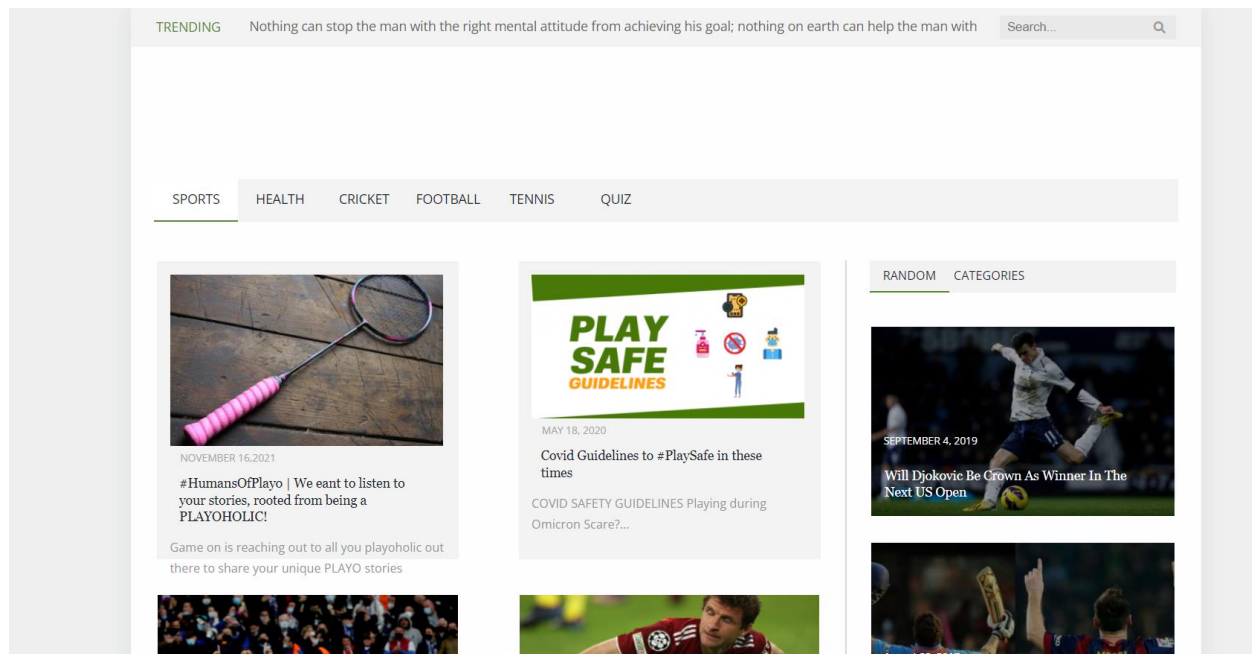
**Racquets**  
Price: \$23/day  
Location: Dallas  
Event Start Date: 2022-04-01  
Event End Date: 2022-04-07







- Blog page with a search feature



- Admin page to add a new venue information into the venues database.

## ADMIN PAGE

img
city name
name
location name
timing
image2
overview
goals
services
infrastructure
address

Add Data

id
img
city name
name
location name
timing
image2
overview
goals
services
infrastructure
address

Update Data

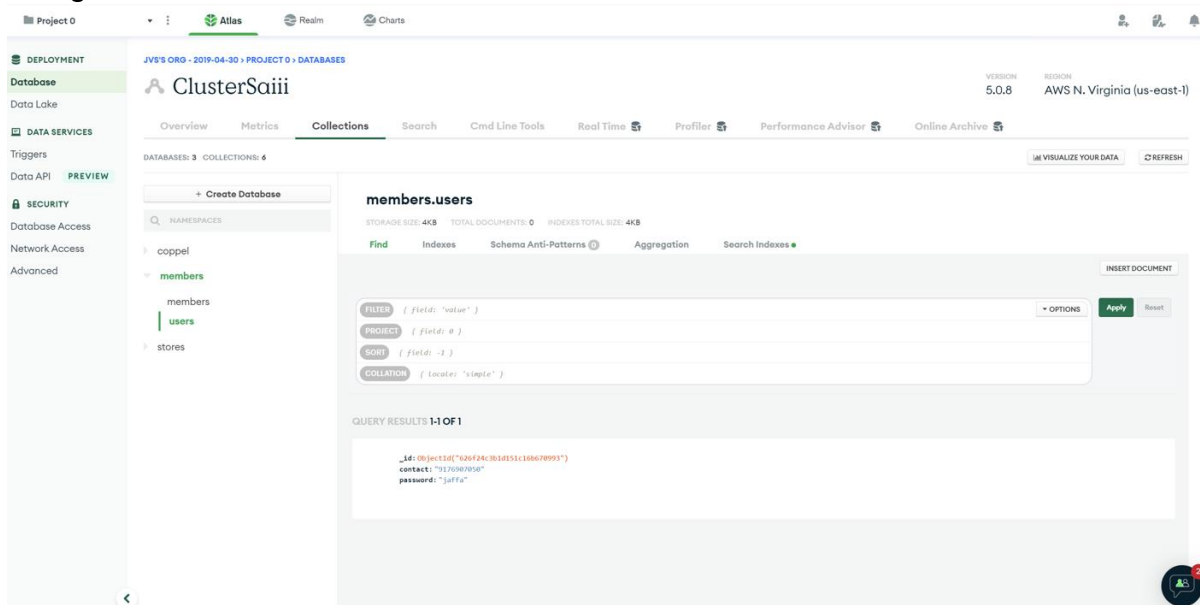
## Data:

- We used a local JSON files to store the data related to the venues and activities. We plan to replace this with TypeORM library

```
[
  {
    "name": "147 One Four Seven Snooker, Billiards and Pool Sports Academy",
    "rating": 0,
    "lat": 12.9341796,
    "lng": 77.6101537,
    "location": "Kansas City",
    "imgUrl": "https://playo.gumlet.io/SPORTSALLEY/SouthUnitedFootballClub1640172478587.jpeg?auto=compress%2Cformat&fit=clip&q=30&f",
    "filter_by": [
      "Pool"
    ],
    "info": "\n    <h3>147 One Four Seven Snooker, Billiards and Pool Sports Academy</h3>\n    <strong>Ratings:</strong> None [N/A]"
  },
  {
    "ratingCount": 6,
    "name": "360 Football Arena",
    "lat": 12.871859,
    "lng": 77.615578,
    "rating": 5,
    "location": "Kansas City",
    "imgUrl": "https://playo.gumlet.io/TIENTO/tientosports6.jpg?auto=compress%2Cformat&fit=clip&q=30&h=0.4&dpr=1&w=1946",
    "filter_by": [
      "Football"
    ],
    "info": "\n    <h3>360 Football Arena</h3>\n    <strong>Ratings:</strong> 5 [6]<br/>\n    <strong>Sports:</strong> Football<br/>"
  }
]
```

```
{
  "city": "hyderabad",
  "name": "Daniel Cricket Academy",
  "location": "hyderabad",
  "image": "images/DanielsCricketAcademycover1502371285385.png",
  "timing": " 6 AM - 8 AM, 4 PM - 8 PM ",
  "id": 1,
  "image2": "images/DanielsCricketAcademycover1502371285385.png",
  "overview": "DANIEL KANAPARTHI, a Level II certified Coach from England and Wales Cricket Board (ECB, United Kingdom).",
  "goals": " To maintain quality coaching standards and contribute best cricket players to represent INDIA. Identifying t",
  "services": "Coaching starts at 5:45 a.m. continues till 8:00 a.m. and in the evening starts at 4:00 p.m. continues ti",
},
{
  "city": "hyderabad",
  "name": "SVM Bowling Alley Gachibowli",
  "location": "36/A, Sai Tirumala Platinum Building, Old Mumbai Hwy, Telecom Nagar Extension",
  "image": "images/Bowling alley.jpg",
  "timing": "All Days | 10 AM - 10 PM",
  "image2": "images/Bowling alley.jpg",
  "overview": "Come Play bowling at SVM Bowling Alley Gachibowli with these amazing offers for all 7 days a week !!",
  "goals": "",
  "services": "Exclusive offers only on Game-On. From (Monday - Thursday) 10 AM - 1 PM - Rs 99/Game 1 PM - 5 PM - Rs 14",
  "infrastructure": "",
  "address": "",
  "id": 2
}
```

## MongoDB database:



## Code Snippets:

- Javascript modules are used to display the common elements across all the inner pages

```
<script type="module">

import footer from "/components/footer.js"

let container = document.querySelector("#footer");
container.innerHTML = footer();

</script>
```

- Utilized localStorage property to permit JavaScript applications to save key-value pairs in an internet browser with no expiry date.

```
import profile from "./scripts/login_profile.js";

var verticalbox2 = document.getElementById("verticalbox2");
verticalbox2.innerHTML = profile()

let n;
var mobnum = JSON.parse(localStorage.getItem("mobileNumber"))

n = document.querySelector("#mobnum").textContent = mobnum;
document.querySelector("#mob").textContent = mobnum;
document.querySelector("#pp3").textContent = mobnum;
```



- Data Related to the venues are stored in JSON format. filtered the user searched venue from the venues database using the Regex expression

```
let response = await fetch("bookvenue.json");
let data = await response.json();
//console.log(data);

//Writing Function for filter data based on search results

let filterData = data.filter((el) => {
  let regex = new RegExp(`^${search_value}`, "gi");
  return el.location.match(regex) || el.name.match(regex);
})
```

- debounce function defers the handling of the keyup event until the user has stopped typing for a considerable amount of time. This prevents our UI code from expecting to deal with each event and furthermore drastically decreases the number of API calls sent off our server.

```
function debouncingFunction(fetchFunction, delay) {

  if (inputTag.value.length == 0) {
    document.querySelector("#show-results-div").style.visibility = 'hidden'
  }

  if (id) {
    clearTimeout(id);
  }

  id = setTimeout(() => {
    fetchFunction()
  }, delay)
}
```

- API calls to fetch news

```
//Api work ==>
let apiKey_blog = "b140754e118c4b63a8e580772d4446a1";
let bloglink = `https://newsapi.org/v2/top-headlines?country=in&category=sports&pageSize=10&apiKey=${apiKey_blog}`;
let response = apiCall(bloglink);
response.then((res)=>{
  console.log(res.articles);
  let appendlocation = document.querySelector(".dynamic-news");
  appendNews(res.articles, appendlocation);
})
.catch((e)=>{
  console.log(e);
})

fetchNews()
```

- Checkout functionality with razor pay

```
var options = {
  "key": "rzp_test_vUKRcpG1a37CW2", // Enter the Key ID generated from the Das
  "amount": price, // Amount is in currency subunits. Default currency is INR.
  "currency": "INR",
  "name": "Game-On",
  "description": "Test Payment",
  "image": "",
  // "order_id": "order_Ef80WJDPBmAeNt", //Pass the `id` obtained in the previ
  // "account_id": "acc_Ef7ArAsdU5t0XL",
  "handler": function (response) {
    alert(response.razorpay_payment_id);
    alert(response.razorpay_order_id);
    alert(response.razorpay_signature)
  }
};
var rzp1 = new Razorpay(options);
document.getElementById('rzp-button1').onclick = function (e) {
  rzp1.open();
  e.preventDefault();
}
</script>
```

- server configuration

```
const PORT = 5000
const path = require('path')
const jsonServer = require('json-server')
const server = jsonServer.create()
const router = jsonServer.router(path.join(__dirname, 'data.json'))
const middlewares = jsonServer.defaults()
server.use(middlewares)
server.use(jsonServer.bodyParser)
server.use('/playo', router)
server.listen(PORT, () => console.log(`JSON Server is running on port ${PORT}`))
```

```
const express = require("express");
const mongoose = require("mongoose");
const users = require("../scripts/loginRoutes");
const bodyParser = require("body-parser");
var cors = require("cors");
const corsOptions = {
  origin: '**',
  credentials: true, //access-control-allow-credentials:true
  optionSuccessStatus: 200,
}
const app = express();
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(cors(corsOptions));
const db = "mongodb+srv://saiii:jvs<3gag@clustersaiii-iacxx.mongodb.net/members?retryWrites=true"

mongoose
  .connect(db, { useNewUrlParser: true })
  .then(() => console.log("Mongodb connected successfully"))
  .catch(err => console.log(err));
//routes use
app.use("/api/users", users);
const host = "0.0.0.0";
const port = 5002;
console.log("deployed on the port: " + port);
app.listen(port, () => console.log(`server is running on port ${port}`));
```

- Filtering method to filter the search based on the selected type of sport

```
//Main Function for filter
function filterFunction(value, div, text){
  console.log(value);
  let filter_array = data.filter((el)=>{
    let array_value = el.filter_by[0];
    //console.log(array_value)
    if(array_value == value){
      return el;
    }
  })

  if(filter_array == undefined || filter_array == [] || filter_array.length == 0){
    document.querySelector(".filterbox").style.visibility = "hidden";
    div.style.border = "2px solid transparent";
    text.style.color = "black";
    let myData = [];
    displayUi(myData)
    return
  }

  console.log(filter_array)

  displayUi(filter_array)
  document.querySelector(".filterbox").style.visibility = "hidden";
  div.style.border = "2px solid transparent";
  text.style.color = "black";
}
```

- Function to fetch venue details based on location

```
async function add(url) {
  try {
    var url = `http://127.0.0.1:3000/cities`;
    let response = await fetch(url, {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
      },
    });
    let d = await response.json();
    console.log("d:", d);
    display(d);
  } catch (err) {
    console.log("err:", err);
  }
}
```

- Activity creation and storing it in the Json file

```
<script>
function submitForm(){
  var location = document.getElementById("location").value;
  var eventName = document.getElementById("eventName").value;
  var eventDate = document.getElementById("eventDate").value;
  var eventTime = document.getElementById("eventTime").value;

  var final = {
    "location":location,
    "eventName":eventName,
    "eventDate":eventDate,
    "eventTime":eventTime,
    "eventBackgroundImage":`https://images.unsplash.com/photo-1585822754398-04873d4e1f50?ixlib=rb-1.2.1&ixid=Mmwo`
  }
  console.log(final)
  let xhr = new XMLHttpRequest()

  xhr.open('POST', 'http://localhost:5001/api/activity/createActivity', true)
  xhr.setRequestHeader('Content-type', 'application/json; charset=UTF-8')
  xhr.send(JSON.stringify(final));

  window.location.href=`http://127.0.0.1:5502/activities.html`
}
</script>
```

- User feedback is received with Formspree feature. It is a backend form, API and email administration for HTML structures. It is the simplest way to add custom contact us forms to your website

```

<div class="right">
  <div id="userinfo">
    <p id="formhead">WRITE TO US</p>
    <input id="uname" type="text" placeholder="Name*" /><br />
    <!-- <input id="uemial" type="email" placeholder="Email*" /><br> -->
    <input id="unumber" type="number" placeholder="Mobile Number" /><br />
    <!-- <textarea id="umessege" placeholder="messege"></textarea>
        <button>SEND</button> -->
    <form action="https://formspree.io/f/xnqwjlzg" method="POST">
      <input
        id="uuemial"
        type="email"
        name="_replyto"
        placeholder="Email*"
      />
      <textarea style="height: 100px" name="message"></textarea>
      <button style="height: 40px" type="submit">SEND</button>
    </form>
  </div>
</div>

```

- Plus and Minus functions to calculate the amount to be paid based on the chosen hours

```

var count = 1;
var costvalue;
function plus() {
  count++;
  if (count > 5) {
    alert("We Dont offer More than 5 hrs");
    return false;
  }

  let plus = document.getElementById("plus");
  plus.style.backgroundColor = "#4daa0b";
  plus.style.color = "white";
  let minus = document.getElementById("minus");
  minus.style.backgroundColor = "white";
  minus.style.color = "grey"

  let updatedvalue = document.getElementById("timevalue");
  updatedvalue.innerHTML = count + " " + "hr";

  costvalue = count * 22;
  cost.innerHTML = "Total cost for these booking - INR" + " " + costvalue;

  return costvalue;
}

```

```

function minus(costvalue) {
  if (count == 1) {
    return false
  }
  count--;
  costvalue = 22 * count;
  let minus = document.getElementById("minus");
  minus.style.backgroundColor = "#4daa0b";
  minus.style.color = "white";
  let plus = document.getElementById("plus");
  plus.style.backgroundColor = "white";
  plus.style.color = "grey";

  let updatedvalue = document.getElementById("timevalue");
  updatedvalue.innerHTML = count + " " + "hr";

  cost.innerHTML = "Total cost for these booking - INR" + " " + costvalue;
  //Let presentValue = costvalue;

}
// below functions for changing the bg color and color of court names
var cost = document.getElementById("cost");

```

## UserSchema:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const UserSchema = new Schema({
  name: {
    type: String
  },
  contact: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  avatar: {
    type: String
  },
  date: {
    type: Date,
    default: Date.now
  }
});

module.exports = User = mongoose.model('users', UserSchema);
```

define routing using methods of the Express app object that correspond to HTTP method

```
// create router
const express = require("express");
const router = express.Router();
const User = require("../User");

router.post("/login", (req, res) => {
  const email = req.body.email;
  const password = req.body.password;
  User.findOne({ contact }).then(user => {
    //user check
    if (!user) {
      errors.contact = "User not found";
      return res.status(404).json(errors);
    }

    //user match
    const payload = { id: user.id, contact: user.contact }; // jwt payload
    console.log("payload", payload);
    return res;
  });
});

//register user
```

```

router.post("/register", (req, res) => {
  const { errors, isValid } = validateRegisterInput(req.body);
  //validation check
  if (!isValid) {
    return res.status(400).json(errors);
  }
  User.findOne({ email: req.body.email }).then(user => {
    if (user) {
      errors.email = "Email already exists...";
      return res.status(400).json(errors);
    } else {
      const newUser = new User({
        name: req.body.name,
        email: req.body.email,
        password: req.body.password
      });

      if (err) throw err;
      newUser
        .save()
        .then(user => res.json(user))
        .catch(err => console.log(err));
    }
  });
});

```

### **Work Sharing/Module Sharing:**

Golla Ranga Kousik/ Nagula Ajay Kumar – End to end Implementation of Book venues, Activities, Blog modules

Srinivas/Manisha – Implementation and integration of Lend module and create Activity features

### **GitHub Link:**

<https://github.com/gollakousik123/WebMobile-2022>

### **Presentation link:**

<https://github.com/gollakousik123/WebMobile-2022/tree/main/PPT>

### **Video Link:**

<https://github.com/gollakousik123/WebMobile-2022/tree/main/Video>

### **References:**

1. Champeon, S. (2003, Mar 21). Progressive enhancement and the future of web design.
2. Retrieved from <http://www.hesketh.com/thought-leadership/ourpublications/progressive-enhancement-and-future-web-design>
3. World Wide Web Consortium (W3C). (2012a). CSS techniques for WCAG 2.0.
4. Retrieved from <http://www.w3.org/TR/2013/NOTE-WCAG20-TECHS20130905/css.html>
5. Guha, S. Krishnamurthi, and T. Jim. Static analysis for Ajax intrusion detection. In International World Wide Web Conference, 2009
6. <https://newsapi.org/docs/get-started>
7. <https://medium.com/codingthesmartway-com-blog/create-a-rest-api-with-json-server-36da8680136d>
8. <https://nodejs.dev/learn>
9. <https://www.meetup.com/>
10. <https://expressjs.com/en/guide/routing.html>