



ATAL BIHARI VAJPAYEE INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR

INTEGRATED POST GRADUATE (IPG) MASTER OF TECHNOLOGY IN
INFORMATION TECHNOLOGY

Minor Project Final Project Report

COMMUNITY DRIVEN SPORTS EVENT
ANNOUNCEMENT AND BLOG POST- FULL STACK
WEBSITE

NAME : ANURAG SRIVASTAVA
ROLL NO. : 2018IMT-019

Under the supervision of
Dr. Kiran Kumar Pattanaik

Contents

1 INTRODUCTION	2
1.1 ABSTRACT	2
1.2 MOTIVATION	2
1.3 LITERATURE SURVEY	2
1.4 PROJECT OBJECTIVES AND FEATURES	3
2 SYSTEM ARCHITECTURE	4
3 DATABASE MANAGEMENT AND DATA MODELLING	5
4 METHODOLOGY	6
5 IMPLEMENTATION	7
6 BLOCK DESIGN DIAGRAM	10
7 TOOLS AND LIBRARIES	10
8 OUTCOME	12
9 CONCLUSION	16
9.1 ADVANTAGES	16
9.2 LIMITATIONS	16
10 FUTURE WORK	16

1 INTRODUCTION

The project creates a platform where players can view and host an event information primarily aimed to increase local level participation and engagement. Also a dedicated blog Post is implemented so that community can interact in the best possible way.

1.1 ABSTRACT

Sports events generally involve elite players and are large-scale. This is perfect for competition among best of bests. But this does little to expose the general public to sports and competition in particular. This project aims to promote the sport and community from local to large-scale events/shows and competitions by providing a platform for organizers, spectators, and participants to connect. Not to mention how important role sports play in a person's life as a skill and as a tool to stay fit.

1.2 MOTIVATION

Often a casual player who obviously doesn't want to pursue the sport professionally is unable to make progress and the primary reason is motivation and knowledge. So, there is a need for a motivating factor and a source of knowledge. The motivation can be brought via competitions and knowledge via the community [4]. Now the only thing that is unavailable is a platform for these two things to thrive. This serves as the background and motivation for the project here.

1.3 LITERATURE SURVEY

Armbet

One of the recent such examples is armbet.com which provides a platform for people to compete with arm wrestlers near them and also host their event over there. The website is in its initial stages, and most of the features are yet to come.

This is one of the motivating factors for this project; here, I aim not only at one sport rather multiple sports and a community blog to help each other.

Codeforces

The blog-post motivation comes from a famous website codeforces.com which has one of the best blog-post systems. The motivation of features like posting a blog, up-voting/down-voting a blog, comment under a blog come from this website.

Sportsjournal

This provides insights into how competitiveness in sports can be beneficial. The competitive environment is reported to influence greater exercise intensity in most persons, thus enhancing practice and training.

1.4 PROJECT OBJECTIVES AND FEATURES

The primary objectives of the projects are as follows

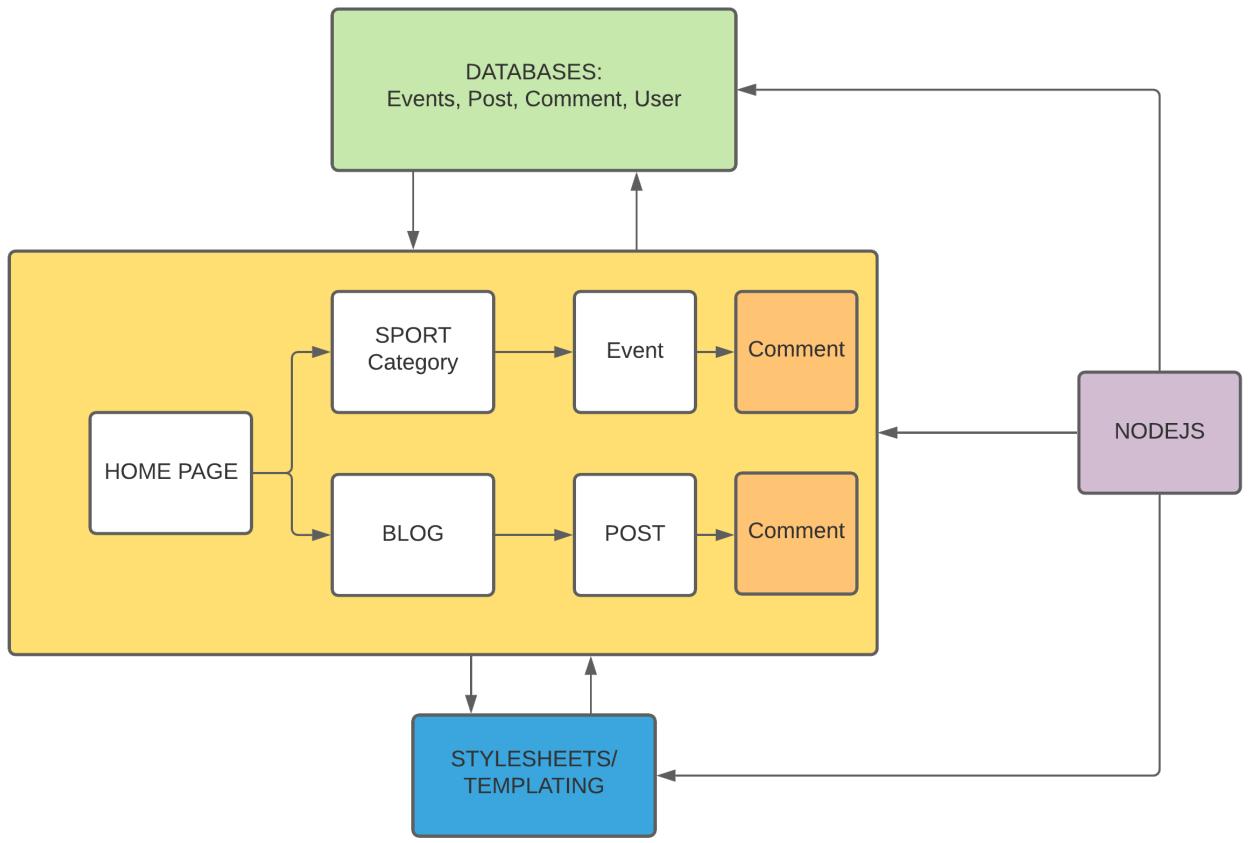
1. Website contains events under the sport category and Blog Post for community discussion. All the content is community driven.
2. Creation of an event under particular Sport Category specifying location and details of event. Location will be shown in the map.
3. Creation of Post under Blog section, users can upload image in an event or blog.
4. Users can comment under a blog post or an event, and are also allowed to up-vote or down-vote comments, post and events.

2 SYSTEM ARCHITECTURE

The is a full-stack website that covers front-end, back-end, and deployment.

1. **Structure:** The structure of the web-page forms the foundation and is called the content layer. This basically comprises of HTML codes. Different parts of a website can be seen within this structure like hyperlinks to navigate throughout the site, the text on the pages, and images that hold it all together.
2. **Style:** The style layer of web design is the completed interior of the site. This is the part of your site visitors see. The structure layer is all of the behind-the-scenes work, and the style makes it look pretty and hospitable. One major complaint of this layer is the Cascading Style Sheets or CSS. This is a document that determines how the site is displayed on a screen, whether it's a large desktop computer, a tablet, or a mobile device. Bootstrap is used primarily for styling.
3. **Behavior:** The final stage of web design is the behavior of the page. This would be the actual functioning of the home, to continue to use our house building metaphor. Here we test how each page interacts with one another and determine how the site reacts to the user's actions on the page. Knowledge of JavaScript, CGI, or PHP is crucial for web developers in this phase.

Databases and nodejs form a behavior layer, stylesheets/Templating are style layers, and Homepage region shows Structure layer.



3 DATABASE MANAGEMENT AND DATA MODELLING

MongoDB [2] is being used for database management. MongoDB is a document-oriented NoSQL [1] database used for high-volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and functions which is the equivalent of relational database tables.

Collections used:

- User- Stores user information like name, email, voted array.

-
- Category- This denotes a category of a sport like swimming, tennis, etc.
 - Event- Stores information of an event which is created by the user containing information like title, location, vote-count, organizer, announcement.
 - Comment- This can be under blog-post or an event which stores author, content, vote-count.
 - Blogpost- A user-created post containing title, author, content, comments.

```
{
  _id: <ObjectId>,
  fullname: String,
  email: String,
  voted: [ObjectId] 
}
```

**Reference to
ObjectId of
already voted
event/comment/
blog-Post**

Above shows an example of ‘user’ collection. Field ‘voted’ stores ObjectId of the event/comment/blog-post that the user has already voted (up-vote/down-vote), so that the user can not vote multiple times on the same object.

4 METHODOLOGY

The main methods of the websites are as follows:

1. Every user is uniquely identified by their email account. Login is done via email and password.
2. The user can view the events under a particular sport category. A registered user can also create an event under the desired sports category.
3. The same is applied to the blog-post. A registered user can create, read, and comment under a blog. A dedicated page is there for every event, blog, and sport category. To create an event or blog a form will be there at the client end.
4. Comment feature can be used by an authorized logged in user who can create a comment under an event/Post and also upvote/downvote any comment, post or event.

-
5. Users creating an event have to select the location of the event using the mapbox given.

5 IMPLEMENTATION

1. vscode text editor is used for writing the code. Visual Studio Code is a free-ware source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
2. For the registration of the user, we will be using a form that asks for name, email, password. The email will serve for verification and further contact purpose. A verification Code/Link will be sent to user mail that will serve as verification. The password stored is hashed using bcrypt.

A bcrypt hash string is of the form:

```
$2b$[cost]$[22 character salt][31 character hash]
```

For example:

```
$2a$10$N9qo8uLoickgx2ZMRZoMyeIjZAfcfl7p92ldGxad68LJZdL17lhWY  
\_/\_ \_/_ /  
Alg Cost Salt Hash
```

Where:

- `$2a$` : The hash algorithm identifier (bcrypt)
- `10` : Cost factor ($2^{10} ==> 1,024$ rounds)
- `N9qo8uLoickgx2ZMRZoMye` : 16-byte (128-bit) salt, base64-encoded to 22 characters
- `IjZAfcfl7p92ldGxad68LJZdL17lhwy` : 24-byte (192-bit) hash, base64-encoded to 31 characters

3. Registration is done in two steps. Firstly User is asked for username and email. An error is displayed if an user already exists with that username or/and email. A verification link is sent to email using mailgun. The link contains token created by jsonwebtoken which contains hashed username and email. Then second page of registration comes that asks for Full name and password.

-
4. The user logged in stays in the browser session up to a certain time period. This is all managed using express-sessions. A currentuser variable is used to determine whether a user is logged in or not so that routes that are to be visited only by logged in users like create/delete blog-post/event won't be accessed by an unauthorized person.
 5. To create an event, the user will fill the form, which includes title, content, image, and location. For location, users can input the location or pin it in the map.
 6. Each blogpost/event/comment has an upvote and downvote button. Upvote increases the vote-count of objects by one and downvote decreases the vote count by one. A user can only either upvote or downvote an object only once. To implement this feature user contains an array of objectID of blog/event for which the user has already voted. Since the vote can only be counted once, If the user tries to vote more than once, the array can be checked for the existence of the Object ID.
 7. All of the database changes are managed by mongoose which is a node-module and serves object modeling for MongoDB.
 8. Deployment is done on Heroku where API KEY of cloudinary, MongoDB URL, and all other environment variables are stored.

Below Code is the entry point of the application. Along with that file structuring is also shown.

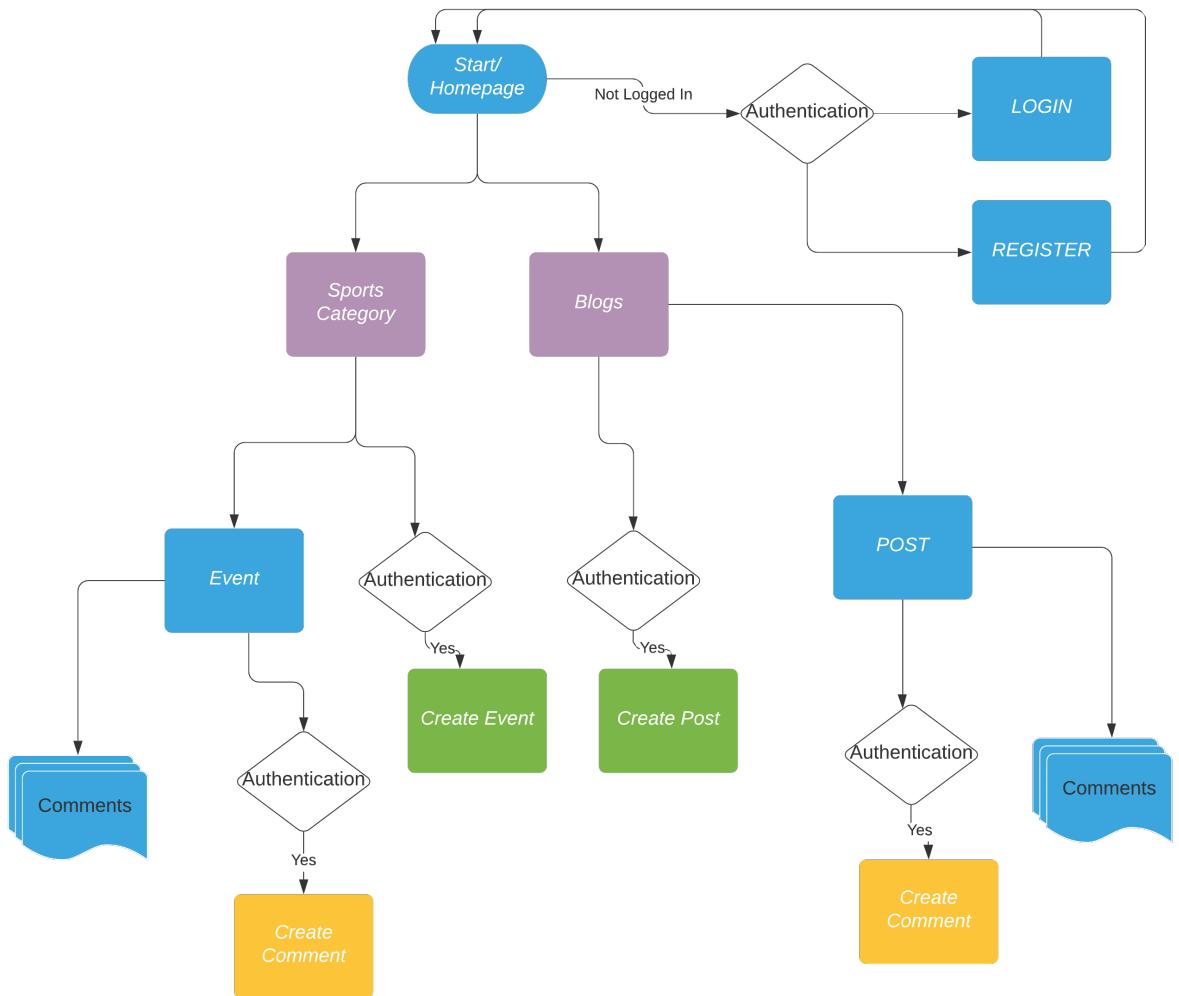
```

1 if(process.env.NODE_ENV !== "production"){
2     require('dotenv').config();
3 }
4
5 const express = require('express');
6 const path = require('path');
7 const ejsMate = require('ejs-mate');
8 const mongoose = require('mongoose');
9 const User = require('../models/user');
10 const passport = require('passport');
11 const LocalStrategy = require('passport-local');
12 const session = require('express-session');
13 const ExpressError = require('../utils/ExpressError');
14 const flash = require('connect-flash');
15 const cors = require('cors');
16
17 const app = express();
18
19 const userRoutes = require('../routes/users');
20 const blogRoutes = require('../routes/blogs');
21 const eventRoutes = require('../routes/events');
22
23 const port = process.env.PORT;
24 const dbUrl = process.env.DB_URL;
25 const secret = process.env.SECRET;
26
27 app.engine('ejs', ejsMate);
28 app.set('view engine', 'ejs');
29 app.set('views', path.join(__dirname, 'views'));
30 app.use(express.urlencoded({ extended: true }));
31 app.use(express.static(path.join(__dirname, 'public')));
32 app.use(cors());
33
34 const sessionConfig = {
35     secret,
36     resave:false,
37     saveUninitialized: true,
38     cookie: {
39         httpOnly:true,
40         expires: Date.now() + 1000*60*60*24*7,
41         maxAge: 1000*60*60*24*7
42     }
43 };
44
45 app.use(session(sessionConfig));
46 app.use(flash());
47 app.use(passport.initialize());
48 app.use(passport.session()); //session should be used before this
49 passport.use(new LocalStrategy(User.authenticate()));
50
51 passport.serializeUser(User.serializeUser());
52 passport.deserializeUser(User.deserializeUser());
53
54 app.use((req,res,next)=>{
55     res.locals.currentuser = req.user;
56     res.locals.success = req.flash('success');
57     res.locals.error = req.flash('error');
58     next();
59 });
60
61 mongoose.connect(dbUrl, {
62     useNewUrlParser: true,
63     useCreateIndex: true,
64     useUnifiedTopology: true,
65     useFindAndModify: false
66 });
67
68 const db = mongoose.connection;
69 db.on("error", console.error.bind(console, "connection error:"));
70 db.once("open", ()=>{
71     console.log("Database connected");
72 });
73
74 app.use('/userRoutes');
75 app.use('/blogRoutes');
76 app.use('/eventRoutes');
77
78 app.get('/',(req,res)=>{
79     res.render('home');
80 });
81
82 app.get('/secret',(req,res)=>{
83     if(req.isAuthenticated()==false){
84         res.send("you must be signed in");
85     }
86     res.send("here is the secret");
87 })
88
89 app.all('*',(req,res,next)=>{
90     next(new ExpressError('Page Not Found',404));
91 });
92
93 app.use((err,req,res,next)=>{
94     const {statusCode}= err;
95     if(!err.message) err.message = 'Something went Wrong';
96     res.status(statusCode).render('error',{err});
97 })
98
99 app.listen(port, ()=>{
100     console.log(`Serving on ${port}`);
101 })

```

controllers
 JS blogs.js
 JS events.js
 JS users.js
 models
 JS blog.js
 JS category.js
 JS comment.js
 JS event.js
 JS user.js
 node_modules
 public
 javascripts
 JS validateForms.js
 stylesheets
 # home.css
 routes
 JS blogs.js
 JS events.js
 JS users.js
 utils
 JS catchAsync.js
 JS ExpressError.js
 views
 blogs
 <> index.ejs
 <> new.ejs
 <> show.ejs
 events
 <> index.ejs
 <> new.ejs
 <> show.ejs
 layouts
 <> boilerplate.ejs
 partials
 <> flash.ejs
 <> footer.ejs
 <> navbar.ejs
 users
 <> auth.ejs
 <> login.ejs
 <> register.ejs
 <> error.ejs
 <> home.ejs
 .env
 JS app.js
 JS middleware.js
 {} package-lock.json
 {} package.json

6 BLOCK DESIGN DIAGRAM



7 TOOLS AND LIBRARIES

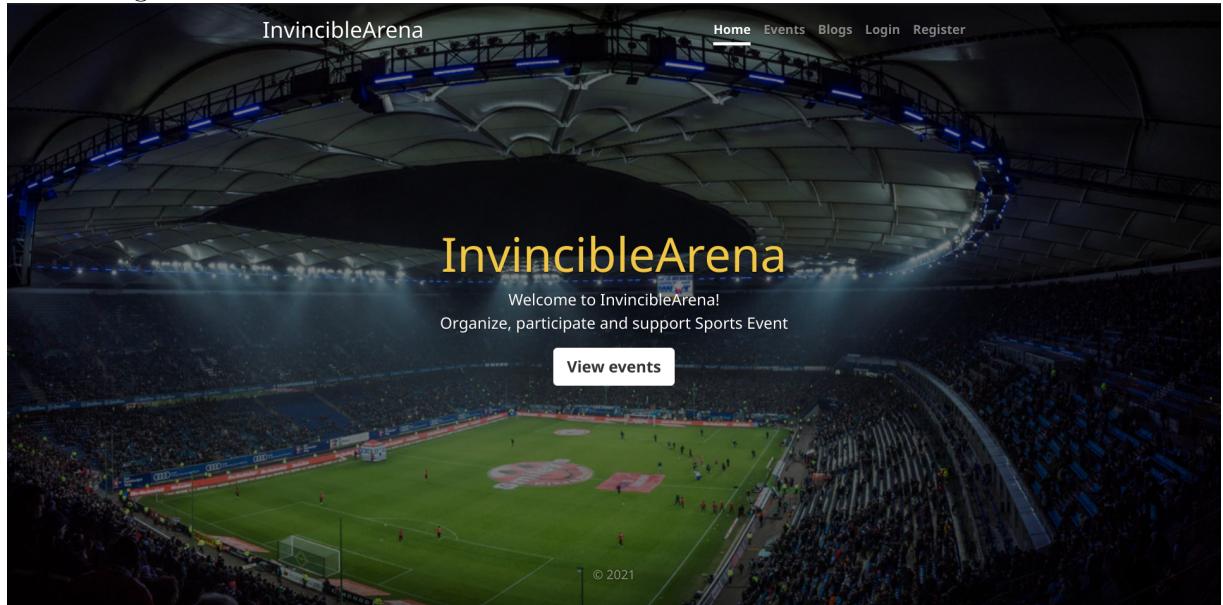
1. Expressjs: Express.js is a free and open-source web application framework for Nodejs [3]. Routes are the most important component of expressjs being used in the project.
2. Ejs: EJS is a template system. You define HTML pages in the EJS syntax and you specify where various data will go on the page. Then, your app combines data with the template and "renders" a complete HTML page

where EJS takes your data and inserts it into the web page according to how you've defined the template.

3. Passportjs: Passport is Express-compatible authentication middleware for Node.js. Passport's sole purpose is to authenticate requests, which it does through an extensible set of plugins known as strategies. Passport does not mount routes or assume any particular database schema, which maximizes flexibility and allows application-level decisions to be made by the developer. The API is simple: you provide Passport a request to authenticate, and Passport provides hooks for controlling what occurs when authentication succeeds or fails.
4. Express-session: Session management is being done in node.js via the express-session module. It helps in saving the data in the key-value form. In this module, the session data is not saved in the cookie itself, just the session ID.
5. Joi: schema description language and data validator for JavaScript.
6. Cloudinary: provides cloud-based video and image management services. It enables users to store, upload, manage, manipulate, and deliver images and videos for websites and apps.
7. Mongoose: Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.
8. Mapbox: It is a JavaScript library that uses WebGL to render interactive maps from vector tiles and Mapbox styles.
9. Helmet: The helmet helps you secure your Express apps by setting various HTTP headers. It provides protection from cross-site scripting (XSS).
10. Dotenv: Dotenv is a zero-dependency module that loads environment variables from a .env file into process.env.
11. Bootstrap: Bootstrap is a CSS-framework. It is free and open-source and contains javascript-based design templates.

8 OUTCOME

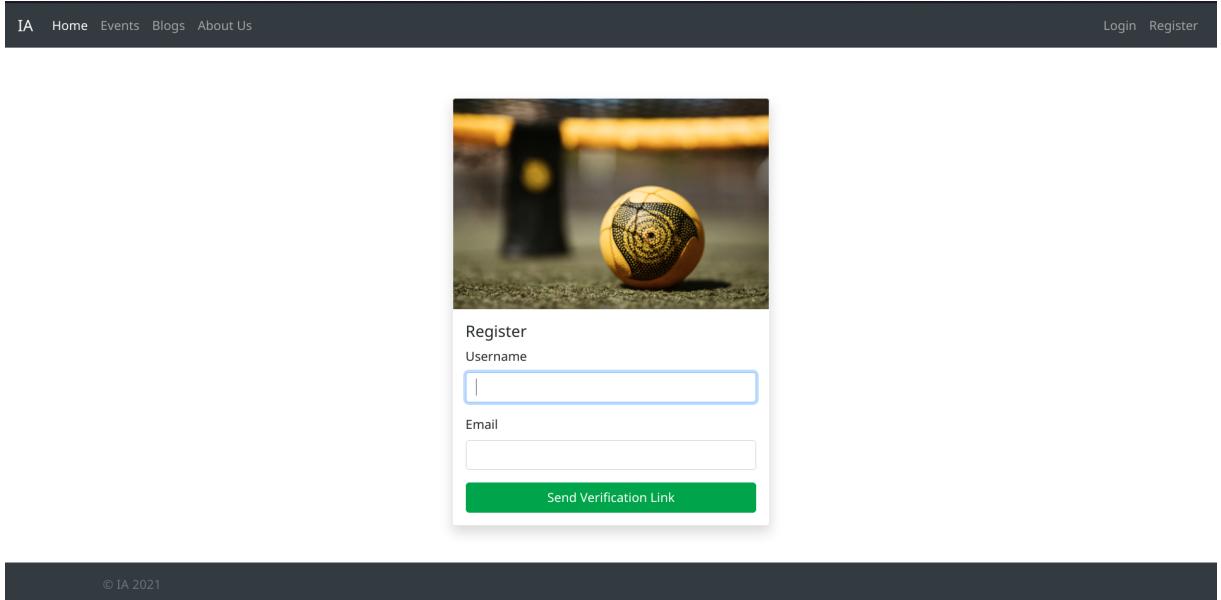
Home Page



Login

The image shows the login page of InvincibleArena. At the top, there is a navigation bar with links for IA, Home, Events, Blogs, About Us, Login, and Register. The main content area features a large image of a soccer ball on a field. Below the image, the word "Login" is displayed, followed by "Username" and a text input field. Below that, the word "Password" is displayed, followed by another text input field. A green "Login" button is positioned at the bottom of the form. At the very bottom of the page, there is a copyright notice: "© IA 2021".

Registration Page to send verification Link



Verification link on mail

Account Activation Link Inbox ×



noreply@invinciblearena.com via sandbox.mgsend.net
to me ▾

Please Click on the given Link to activate you InvincibleArena Account [Click Here](#)

Reply

Forward

Registration After Verification from Link

IA Home Events Blogs About Us
Login Register



Verification

Full Name

Password

Register

© IA 2021

BlogPost Index Page

IA Home Events Blogs About Us
Login Register

Create Blog

Temporary ..

Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint saepe mollitia eveniet praesentium! Saepe natus velit quis sint pariatur sunt quasi preferendis, dolor fugiat nostrum, iure unde totam est distinctio. Tempore repellat adipisci, nesciunt natus eligendi ipsum labore dolor odit autem, alias
— Thu May 13 2021 19:59:31 GMT+0530 (India Standard Time)

[View](#)

Testing, testing... GIFs in chat, following specific threads, ..

GIFs are coming to chat Whether you want to tease a friend, react to something funny, or show your current mood, the ability to share GIFs gives you more to work with while chatting it up with your fellow redditors. Starting today, we're testing allowing redditors on the web, iOS, and Android to sh
— Thu May 13 2021 18:01:34 GMT+0530 (India Standard Time)

[View](#)

© IA 2021

Blog Show Page

Temporary

Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint saepe mollitia eveniet praesentium! Saepe natus velit quis sint pariatur sunt quasi perforendis, dolor fugiat nostrum, iure unde totam est distinctio. Tempore repellat adipisci, nesciunt natus eligendi ipsum labore dolor odit autem, alias eius eveniet ipsam tempora modi incident molestiae iusto, error voluptatem minus ut accusamus quis mollitia. Sapiente quos magni ex voluptates, inventore qui soluta, saepe nesciunt at corrupti natus et distinctio ratione accusantium rerum labore obcaecati ab dolore quasi rem voluptatum sequi explicabo facilis! Adipisci dicta minima impedit vel autem, quidem odit suscipit vitae repudiandae, natus repellendus, esse rerum.

Anurag Srivastava

Thu May 13 2021 19:59:31 GMT+0530 (India Standard Time)

Vote Count: 0



Leave Comment

[Comment](#)

Comments

Anurag Srivastava

nice

Thu May 13 2021 20:00:06 GMT+0530 (India Standard Time)

Vote Count: -1



Event Index Page



Football

[View](#)

Basketball

[View](#)

Armwrestling

[View](#)

Powerlifting

[View](#)

Mapbox



9 CONCLUSION

9.1 ADVANTAGES

- Growth of sports and competitiveness among players.
- Opportunity of self-branding
- An easy blog post serving various queries from an active community.

9.2 LIMITATIONS

- The authenticity of the blog is still up to judgment from vote count and user and is not verified by any means.
- No registration method included for an event as of now.

10 FUTURE WORK

1. Videos of an event after it's over can be included so that those who cannot go there can still see that event.
2. Registration and Payment method can be included for an event.
3. Commercialization of the website: This is the most important thing not only for the website to thrive but also for the community to grow. There are several ways:
 - Products under our brand.

-
- Taking some percentage from users registering for an event/ charges for hosting an event on the website.
 - Paid Telecast/Videos of an event.

References

- [1] Jing Han, Haihong E, Guan Le, and Jian Du. Survey on nosql database. In *2011 6th International Conference on Pervasive Computing and Applications*, pages 363–366, 2011.
- [2] A. Kanade, A. Gopal, and S. Kanade. A study of normalization and embedding in mongodb. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 416–421, 2014.
- [3] Matt Loring, Mark Marron, and Daan Leijen. Semantics of asynchronous javascript. In *Proceedings of the 2017 Symposium on Dynamic Languages*, pages 51–62, 2017.
- [4] Sean Lowton-Smith, Richard Morgan, Michelle Stanley, Thomas Hames, Philip Smith, Chelsey Lawson, and Sheila Leddington Wright. Peer-to-peer teaching: Experience of 3rd year undergraduate sports therapy students and impact upon applied academic performance. *Journal of Hospitality, Leisure, Sport & Tourism Education*, 25:100196, 2019.