
Software Requirements Specification

for

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience
- 1.4 Product Scope
- 1.5 References

2. Overall Description

- 2.1 Product Functions
- 2.2 User Classes and Characteristics
- 2.3 Use Case Diagram and Use Cases
- 2.4 Design and implementation Constraints
- 2.5 User Stories
 - 2.5.1 Viewer's user stories
 - 2.5.2 Track Organizer's User stories
 - 2.5.3 Admin User Stories
 - 2.5.4 Participant User stories
- 2.6 Assumptions and dependencies

3. External Interface Requirements 3

4. System Features 4

5. Other Nonfunctional Requirements 4

1. Introduction

1.1Purpose

Track Management system is a web portal where competition(tracks) or hackathons will be conducted in which people can participate in ongoing tracks and submit their solutions. They will be evaluated on certain metrics and participants will have a leaderboard. The purpose of this project is to create a website which is user-friendly where participants can showcase their problem-solving skills.

1.2 Intended Audience and Reading Suggestions

This document is targeted at developers, testers, users and other documentation writers. The document contains project details, functional and non-functional requirements, User stories, Use cases, Sequence diagrams, features, etc. Here are some reading suggestions for

- Developers: can skip product perspective.
- Testers: relevant topics are functional and non-functional requirements.
- Users: can skip technical aspects and read product scope and product features.

1.3 Product Scope

- The Track Management System will be useful for conducting competitions online with a certain number of participants.
- This site can be used by organizers who want to organize a new competition and also by participants to participate.
- Viewers(non-participants) can also see the ongoing tracks and look at track details.
- In the future, product scope can be expanded by adding features where users can access the data models, datasets, and code.

2. Overall Description

2.1 Product Functions

- Login and sign-up for participant and organizer.
- Verification of user
- Able to add a new track for organizers(competitions).
- Edit track details for organizer.
- Showcase tracks yearwise.
- Chat application.
- Leaderboard.

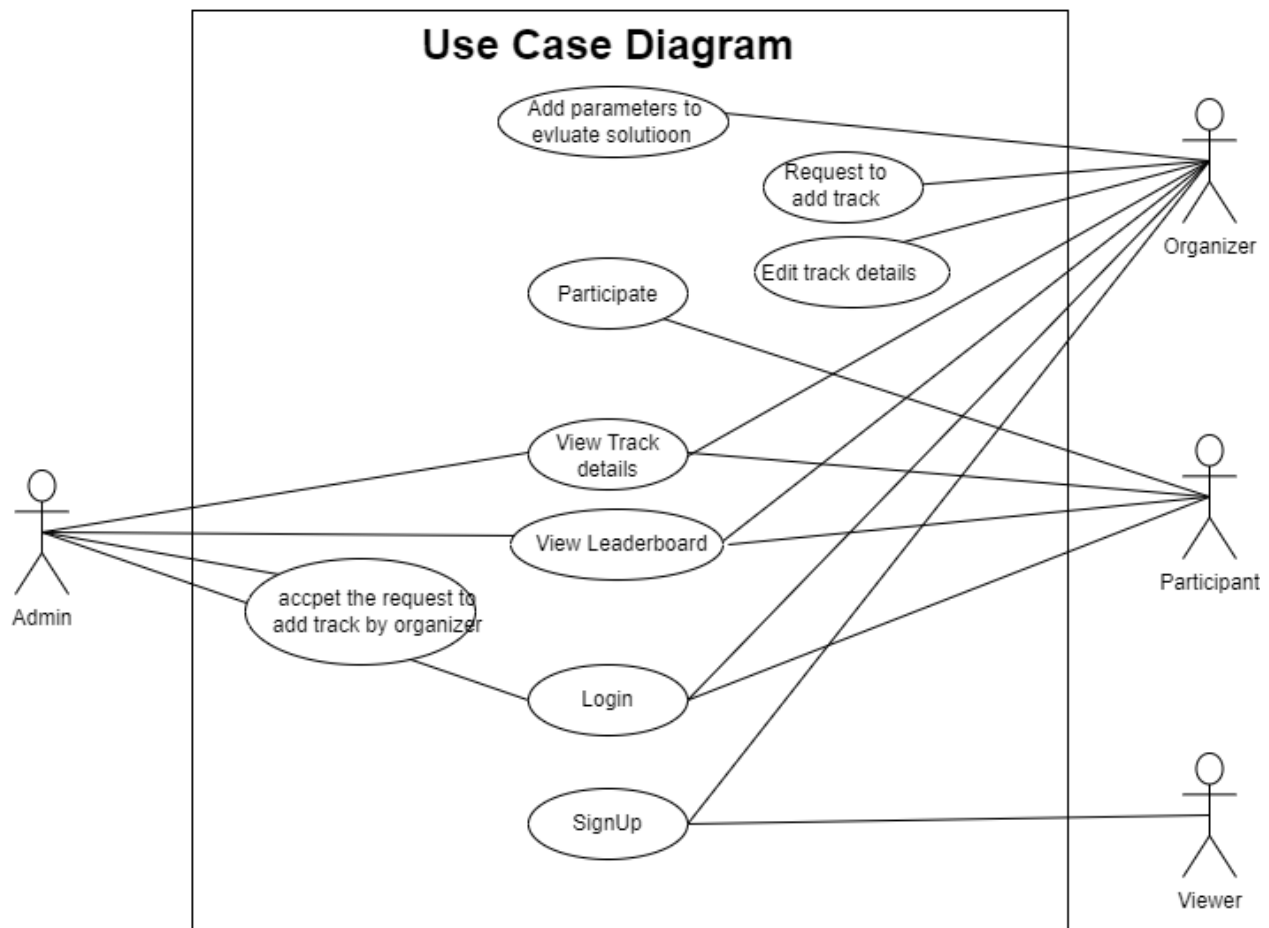
2.2 User Classes and Characteristics

There will be four types of users: Admin,Organizer,Viewer,Participant.

The following are their privileges:

- Viewer: View ongoing tracks and Sign-Up for participation.
- Participant: They can inherit above class privileges and also Login,Participate,View Track details,view Leaderboard.
- Organizer: Excluding participation it can inherit above classes privileges and also can evaluate solutions, edit track details, request to add track.
- Admin: They can inherit above class privileges and also have other privileges like validate Track, Add Track, Accept request from organizer.

2.3 Use Case Diagram and Use cases



Following are the use cases:

- **Participating in competitions:** The website provides a platform for individuals and teams to participate in competitions .
- **Hosting competitions:** Organizers can also host their own competitions on the website and invite participants to compete.
- **Forming teams and collaborating:** The website allows users to form teams and collaborate on machine learning challenges.

- Request to add tracks: Allows organizers to send requests to add new tracks to admin. Afterwards Admin can review the request and add new track to database.
- Editing track details: Allows Organizer to edit track details of any existing tracks.

2.4 Design and Implementation Constraints

We have developed this project using MERN stack which comprises technologies like MongoDB, Express, React, Node.js. This technology provides better integration by giving a strong support of documentation and developer community in case of any problem. Therefore, it is important to strictly adhere to the MERN stack and avoid introducing additional technologies that may not be compatible.

2.5 User Stories

2.5.1 Viewer's User Stories:

1- Front of card:

"As a viewer, I want to have access to the activity calendar so that I can review the timelines."

Back of card:

1. *I will not be able to change the calendars.*

2-Front of card:

“As a viewer, I want to have access to view leaderboards. So that I can analyze participants .”

Back of card:

1. I will not be able to change any details of the track.
2. In case there's no participant then the result will display empty.

3-Front of card:

“As a viewer, I want to have access to search for a person/team name in the leaderboards. So that I can analyze participants .”

Back of card:

1. I will not be able to make changes in the leaderboard.
2. In case there's no participant/team then the result will display empty.

4-Front of card:

“As a viewer, I want to have access to view the homepage of a particular track. So that I can review the track.”

Back of card:

1. I will not be able to change any details of the track.
2. In case there's no participant then the result will display empty.

2.5.2 Track Organizer's user stories

1) Front of card:

“As a Track organizer, I want to add a track, so that I can organize an event”

Back of the card:

1. Capture the track details like track description, and the necessary data sets, duration of contest, start time of contest .
2. Verify that the organizer is signed in.
3. Verify that the admin accepts the request of adding the track.

2) Front of card:

“As a Track organizer, I want to edit track details so that I can make changes to an existing track “

Back of card:

1. Capture the change in track details
2. Verify the organizer is signed in
3. Verify the organizer has access to the track
4. Verify the track exists.

3) Front of card:

“As a track organizer, I want to have access to view leaderboards so that I can analyze the participants”

Back of the card:

1. Verify the organizer is signed in.
2. Verify there exists a track name similar to the requested one.

4) Front of card:

“As a Track organizer ,I want to be able to Sign in so that I can access my account.”

Back of card:

1. Confirmation

Success:

1. “Remember me” ticked : store cookie for automatic login next time.
2. “Remember me” not ticked : Force login next time.

Failure:

1. “Email address format not correct”.
2. “Entered email address is wrong”.
3. “Entered Password is incorrect”.
4. “Not the Track organizer details”.

2.5.3 Admin User stories:

1. Front of card:

“As an Admin, I want to be able to Sign in so that I can access my account.”

Back of card:

Confirmation

Success:

1. “Remember me” ticked : store cookie for automatic login next time.
2. “Remember me” not ticked : Force login next time.

Failure:

1. “Email address format not correct”.
2. “Entered email address is wrong”.
3. “Entered Password is incorrect”.
4. “Not a admin user details”.

2. Front of the card:

“As an admin, I want to be able to manage user accounts, including adding new users, modifying existing accounts,when necessary so that I can keep the user details updated.”

Back of the card:

Success:

1. Add new users if they have valid signing up credentials for new users.
2. They have filled all the required parameters for signing up for new users.
3. “Log in successfully” if they have passed all the requirements.
4. Modify and monitor existing user details such as score and number of events, submissions and progress based on their performance and participation.

Failure:

1. Invalid log-in/sign-in credentials.
2. If the user forgets password then send password reset mail to its email account by using the option of forgot password.

3.) Front of the card:

“As an admin, I want to be able to monitor the performance and health of the platform, including server status ,security and privacy features so that the user doesn’t experience any difficulty.”

Back of the card:

1. Track should be able to handle a significant amount of user traffic.
2. The user data should be stored securely and implement privacy.

7.)Front of the card:

“As an admin, I want to be able to communicate with users through emails and platforms so that I can provide updates, learn from feedback, and support.”

Back of the card:

1. Provides email-ID's on platform for users and organizers to provide feedback and contact for any type of support and guidance to use the platform.
2. Keep basic documentation for usage and solution to common platform issues.
3. Focus on the relevant feedbacks and make necessary changes and improve the platform.

8.)Front of the card:

“As an admin, I want to be able to generate reports and analytics on platform usage, user behavior, and contest performance so that the user and organizer can access it for their learning and helping the platform for future development and decision-making.

Back of the card:

1. Develop data and analytics based on the track event and make it available for the organizers and users.
2. Focus on the platform performance in a live track and make necessary changes for improvements by generating the performance history during a track.

9.) Front of card:

“As an admin, I should be able to delete any track as needed to manage data properly and efficiently.”

Back of card:

Success:

“Track deleted Successfully”

Failure:

Invalid Track ID

10.) Front of Card:

“As an admin, I should be able to update any track details to manage tracks properly”

Back of card

Success :

“Track Details Updated Successfully”

Failure:

Invalid Track id

2.5.4 Participant User Stories:

1. Front of card:

“As a participant, I want to have access to the activity calendar so that I can review the timelines.”

Back of card:

1. I will not be able to change the calendars.

2. Front of card:

“I want a feature where if I lose my password , then I am allowed to login through my trusted device where I had already signed in using some form of passcode to recognize it is a valid account owner attempting to sign in”

Back of card:

Success:

1. If the user enters the correct Username/Email and password.

Failed:

1. User enters a wrong password several times and is presented a way to try other ways to sign in.
2. User is presented with two ways , either use a phone and passcode to sign in or to reset password through email.
3. If a user enters a wrong code , the option to sign in through phone is revoked and then is asked to reset password through email.

3. Front of card:

“As a Participant I want to Login to the platform.”

Back of card:

Success:

1. Participant will be redirect to Home Page of the platform

2. Remember me ticked - store cookies/ next time autologin
3. Remember me not ticked - repeat login next time

4. Front of card:

“I want simple interface and easy navigation for competition’s list”

Back of card:

1. The pages can have a minimalistic background with highlighted buttons so as to draw attention to correct options with correct labeling on them.
2. The competition list and ongoing competitions can be shown on the left hand scrollable tab with its own scroll bar.
3. The leaderboard top 3 participants can be shown on the right tab.

5. Front of card:

“I want a clear and in-detail leaderboard for all participants for a particular competition. what factors are responsible for ranking must be given with deep details.”

Back of card:

1. This can be done when a user clicks on a particular competition , then the user will be redirected to a competition’s landing page where a short description of competition will be displayed and then the page will show the top few participants and user’s current position.

6. Front of card:

“I want a list of all the competitions from the past few years(5-7 years).”

Back of card:

1. A drop down tab named ‘Past competitions’ can be used which will show all competitions from the past 7 years.
2. If the competition is older than that , then users can click on ‘More older competitions’ to see a list of competitions sorted yearwise.

7. Front of card:

“I want all details of prize distribution”

Back of card:

1. A page can be created for each competition which has finished to check if the results have been declared. For this, a link can added named as ‘Standings’ or ‘Results’
2. If the competition is going on then the result will not appear and it will show a timer for remaining time.
3. If the result has not been declared then it will show that the ‘Result has not been declared’ .
4. If the result has been declared then it will show a result page along with prizes which have been distributed to them.

8. Front of card:

“As a viewer, I want to be able to search profiles of registered users on websites so that I can analyze participant progress.”

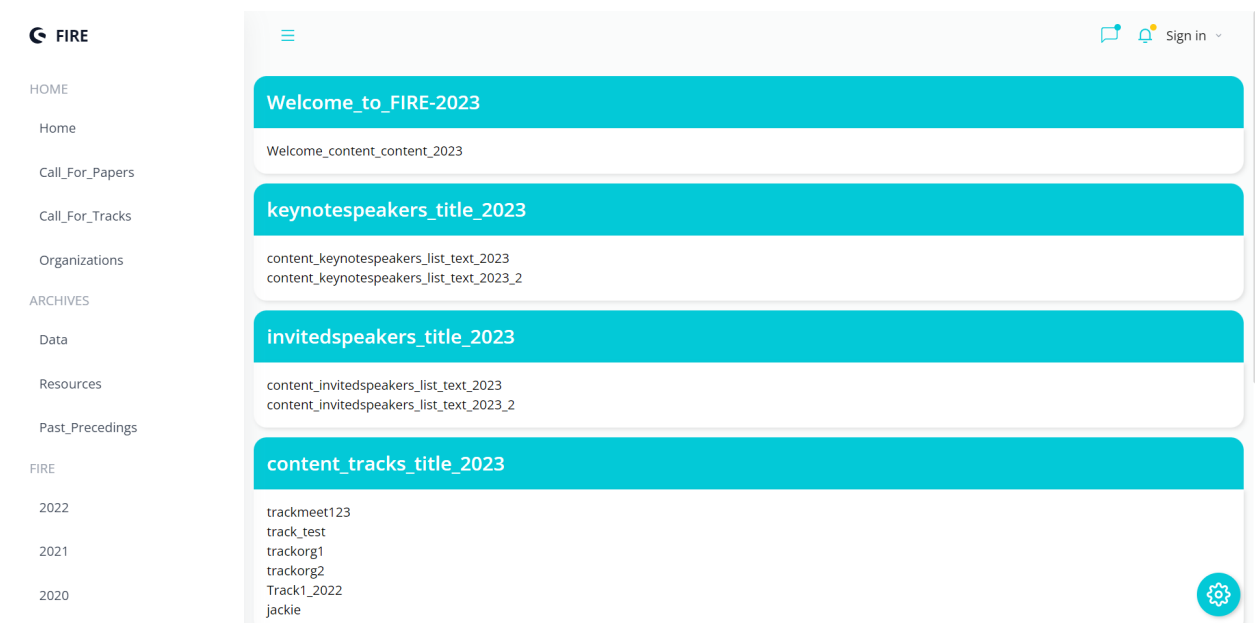
Back of Card:

1. I will not be able to change the profile of an individual.
2. If there's no person matching with the search string then the result will display empty.

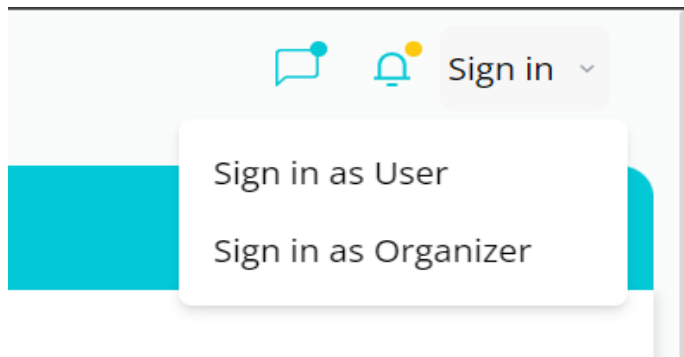
3. External Interface Requirements

3.1 User Interfaces

Home page:




SignIn UI:



SIGN IN

SIGN UP



Sign In for Participant

Username *

Password *

☐ Remember me

SIGN IN


Forgot password ?

Do you have an account ?[Sign Up](#)

SignUp UI:

SIGN IN

SIGN UP



Sign Up For Participant

Please fill this form to create an account !

Name

Email

Gender

☐ Female ☐ Male

Phone Number


Password

Confirm Password

☐ I accept all T&C.

SIGN-UP

Track Details Page:




HOME

Home

TrackDetails

Leaderboard

Submit



Register Team

Introduction

This is content of Introduction.

Task Description

This is content of Task Description

Training Corpus


Content of Training Corpus

Registration


Content of Registration

Submission Format

Content of Submission.



Leaderboard Page:




HOME

Home

TrackDetails

Leaderboard

Submit




Register Team

Leaderboard

Rank	Team Name	Team Score
1	meet123	1500
2	gp2	0
3	meet123456	0
4	meet1234	0

Submit Page:




HOME

Home

TrackDetails

Leaderboard

Submit



Sign in

Upload

Choose File

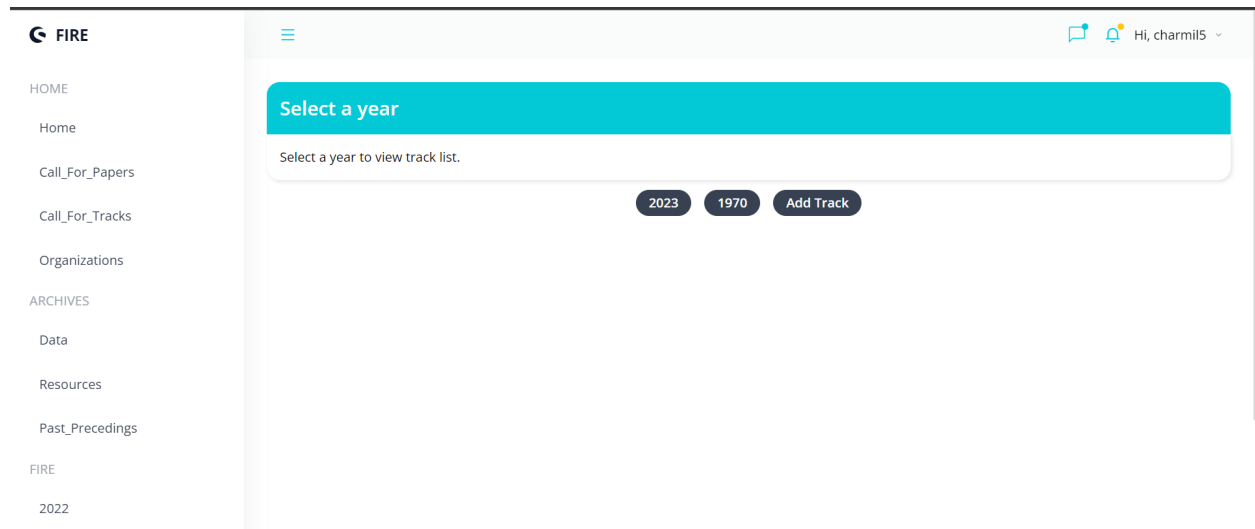
No file chosen

Upload

Download

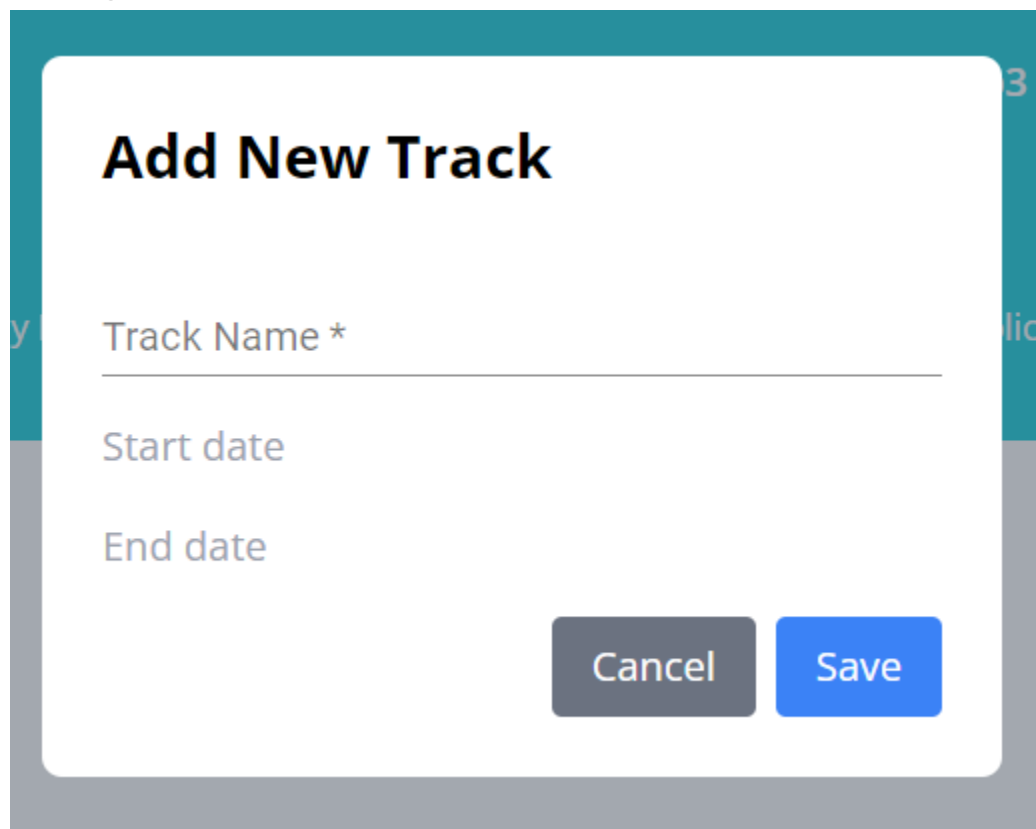
Download

Organizer Page:



The screenshot shows the FIRE Organizer Page. On the left is a sidebar with a 'FIRE' logo and a menu containing: HOME, Home, Call_For_Papers, Call_For_Tracks, Organizations, ARCHIVES, Data, Resources, Past_Precedings, FIRE, and 2022. The main content area has a header with a hamburger menu icon and a user profile 'Hi, charmil5'. Below the header is a teal bar with the text 'Select a year'. Underneath this bar is a white box with the text 'Select a year to view track list.' and three buttons: '2023', '1970', and 'Add Track'.

Adding New Track Details:



The screenshot shows a modal form titled 'Add New Track'. It contains three input fields: 'Track Name *', 'Start date', and 'End date'. At the bottom right of the form are two buttons: 'Cancel' and 'Save'.

Edit Track Details Page:

The screenshot shows a web interface for editing track details. It features a blue header bar with a close button (X) and the title 'Details', and a 'SAVE' button on the right. Below the header are five stacked text input fields, each with a label and a placeholder text:

- Introduction**: This is the Content of Introduction.
- Task description**: Content of Task Description.
- corpus**: Content of Training Corpus
- Registration**: Content of Registration
- Submission**: Content of Submission.

3.2 Hardware Interfaces

Supported devices: The website is designed to be accessed through desktop and laptop computers, as well as mobile devices such as smartphones and tablets. The website is optimized for various screen sizes, making it accessible to a wide range of users.

User interface: The website has a user-friendly interface that allows users to navigate through the website, view information about competitions and research papers, register and log in, and participate in competitions or submit research papers. The interface is designed to be intuitive and easy to use, ensuring a seamless user experience.

Backend technology: The website uses the MERN stack (MongoDB, ExpressJS, ReactJS, and NodeJS) to build the backend and frontend

components of the website. This stack provides a robust and scalable framework for building web applications, ensuring optimal performance and reliability.

3.3 Software Interfaces

1. **Databases** : The website uses MongoDB as the primary database to store all user-related information such as user profiles, competition details, and research papers. The website communicates with the MongoDB database using the Mongoose library, which provides a simple and elegant way to manage data.
2. **Operating System** : The website is designed to be platform-independent and can run on any operating system that supports NodeJS and MongoDB. The website uses NodeJS as the primary runtime environment, which allows the website to run on multiple operating systems such as Windows, macOS, and Linux.
3. **Tools and libraries** : The website uses various open-source libraries such as ExpressJS, ReactJS, Mongoose, and PassportJS. ExpressJS is used as the web application framework, ReactJS is used to build the frontend components, Mongoose is used to manage the MongoDB database, and PassportJS is used for authentication and authorization.

3.4 Communications Interfaces

The communication will be done totally on the webpage and no other means of communication like mailing exists. The communication among the backend and frontend is done through an API and HTTP responses from the user are used to render data.

4. Functional Requirements

1. The Track management system should let users, administrators, and track coordinators log in and have separate login credentials.
2. The Track management system should be able to add a user to a particular track he wishes, given the permission of the track coordinator.
3. The Track management system should be able to add a new track on the request of a coordinator.
4. The Track management system should allow the track coordinator to update the track he owns.
5. The Track management system should be able to add a new user and a Track coordinator and add them into the database.
6. The Track management system should accept team registrations in the track entry portal given the individuals are priorly registered as users.
7. The Track management system should allow users to view the past year's track details and their leaderboards.

5. Other Nonfunctional Requirements

Apart from the functionalities the user can directly experience, there are some other requirements which are related to the behavior of software in certain conditions.

1.Performance: Given the large number of datasets provided to the software and concurrently running multiple tracks, performance is paramount and for a strict competitive environment where each second matters a lot, response time is expected to be less.

2.Scalability: Given the increasing awareness of such softwares and high demand, there will be a definite rise in users and data volumes, hence there is a need to accommodate growing volumes of data without compromising on security and performance.

3.Security: The system should have robust security measures in place to protect user data and sensitive information, such as user authentication, data encryption, and access controls.

4.Usability: Given the wide variety of users , The system should be easy to use and navigate, with an intuitive user interface that allows the administrators and the users to quickly and efficiently access the information they need.

5.Maintainability:The system should be easy to maintain and upgrade, with clear documentation, well-organized code, and support for automated testing and deployment.

6.Reliability:The Server would perform desired tasks as expected. The system does its work with more accuracy like user registration to the system, user validation and authorization, track search and register operation, current leaderboard , and updating the database by synchronizing between database and application.

