Backend Environment Setup

- Extract the contents in the file sportsevent-backend.7z

- Make sure JDK ,JRE and MVN are installed in local environment

- Install STS from <https://spring.io/tools> to run the application as Spring Boot application

- Open the STS Eclipse application and Import the extracted project

- Run ‘mvn install’ for the project to download all the necessary dependency files.

Backend Project setup

- Install Postman, Mysql and mysql workbench in local

- Run the ‘CreateTablesAndSchema’ file in src -> main -> java -> com -> sportsevent -> backend -> database path as a java application.

- After running the application make sure the schema ‘sports\_event’ is created in mysql.

- Also check if the tables ‘user’, ‘event’, ‘participant’ and ‘message’ tables are created in the schema.

- Open mysql workbench and import the ‘event.csv’ and ‘participant.csv’ files to their respective tables

- Now run the project as spring boot application

- Open the postman application and check if the backend Rest endpoints are working by calling the endpoints (Check below for API information)

# Verify user (API to be called when user Signs In)

endpoint – <http://localhost:8080/user/login>

method – POST

requestBody example -

{

“username” : “root”,

“password” : “root”

}

sample response 1(If user accout is registered)

- true

sample response 2(If user accout is not registered)

- false

# Register user (API to be called when user registers)

endpoint – [http://localhost:8080/user](http://localhost:8080/user/login)/register

method – POST

requestBody example -

{

“username” : “root”,

“password” : “root”,

“email” : “[sample@gmail.com](mailto:sample@gmail.com)”,

“phone” : “+19829982891”

}

sample response 1(If user registration is successful)

- true

sample response 2(If user registration is not successful)

- false

# Get EVENTs (API to get all the EVENTS)

endpoint – [http://localhost:8080/s](http://localhost:8080/user/login)ports/events

method – GET

sample response :

{[ {

“id” : 1,

“name” : “Long Jump”,

“Location” : “USA”,

“time” : “9:00 – 10:00”

“date” : “11-10-2021”,

“eventType” : “ONGOING”,

“participants” : [

{“id” : 1,

“name” : “John”,

“eventId” : 1},

{“id” : 2,

“name” : “wick”,

“eventId” : 1}

]},

{

“id” : 2,

“name” : “High Jump”,

“Location” : “USA”,

“time” : “9:00 – 10:00”

“date” : “12-10-2021”,

“eventType” : “UPCOMING”,

“participants” : [

{“id” : 3,

“name” : “David”,

“eventId” : 2},

{“id” : 3,

“name” : “Charles”,

“eventId” : 2}

]}

]}

# Get Messages (API to get all the Messages for discussion forum)

endpoint – [http://localhost:8080/](http://localhost:8080/user/login)forum/discussions

method – GET

sample response -

{[

{

“id” : 1,

“text” : “Hello”,

“userId” : 1,

“time” : “2021-10-11 20:35:43.110”,

“username” : “John”

},

{

“id” : 2,

“text” : “Good morning”,

“userId” : 2,

“time” : “2021-10-11 20:36:43.110”,

“username” : “Wick”

}

]}

# Get Messages (API to get all the Messages for discussion forum)

endpoint – [http://localhost:8080/](http://localhost:8080/user/login)forum/discussions

method – GET

sample RequestBody -

{

“text” : “I would like to know the timing of the High Jump event”,

“userId” : 1

}

sample response 1(If message is saved in DB successfully)

- true

sample response 2(If message is not saved in DB successfully)

- false