**Sesha: General Comments:**

1. **Building all access through athleteID and hiding PhoneNumbers.**

    1. **Send Sign-in as HTTPS with phone number and other profile as part of the body. Phone number will be encrypted in the body.**

    2. **Cloud will create a table of (Phone-number (key) and generated athleteId)**

    3. **Return the generated athleteId to the client as part of HTTPS response.**

    4. **The primary athlete document will have athleteId as the primary Key for access.**

**Android Cognito module uses TLS(advanced SSL) to handshake with Cognito pool running on AWS server.**

**During signup Mita-Ath app captures  mobile number , email id , given name , password , sent by secure webservice call , will insert data into pool  and returns sucess with valid session id or token.**

**Cognito provides APIs , Create custom attribute called "ATHLETE_ID" in the  Athelete pool. When a new athlete is created on the Cognito, insert a new record in the mongoDB , update ATHLETE_ID on the pool using Cognito APIs**

2. **Similar approach should be followed for CoachID**

    **Ps refer above details**

3. **There are quite a few English errors in the API and it sort of makes it not easy to search.**

4. **Avoiding IP address in the API's. Can the client do mitafitness.in/<rest of the API> ? I understand we did not have domain name initially. One way to do this is the following.**

    1. **In your properties files have a field called mitafitness.in = <IPaddress>**

    2. **If this property does not exist, then form the URL as mitafitness.in/API**

    3. **If this property exists, replace mitafitness.in by the IP address mentioned in the property file.**

**Yes , IP address are hard coded now , best practise are as follows**

1. **www.mitafitness.in and mitafitness.in can points to your static web page contents**

2. **Create sub domain like app.Mitafitness.in or www.mitafitness.in/app can point to application webservice calls**

3. **Install SSL certificate to run it on the 443 port , which is safe and secure.**

**Steps 2 already implemented on 31-07-2020**

5. **I thought tomcat will provide a port 80 based usage and we can have some php scripts to map to corresponding microservice calls. Why port 85xx port was used in API ?**

**Yes It is possible run tomcat app server on port 80 by running “*authbind*” *, but we don’t recomend it . It opens door for aubse or hack scenarios.***

***Instead use the following best practices ,***

1. ***You can run apache httpd web server on port 80 or 443 secure , it acts as proxy server to tomcat server, safe also.***

2. ***Your web site static content can be deployed on apache httpd server.***

3. ***As you said you have some PHP scripts, Python scripts , easily these scripts can be deployed on Apache httpd server using bridge modules.***

4. ***Down the line if you want to scale horizontally then it is easy for apache httpd server to handle as load balancer and have multiple tomcat servers running on the back end.***

5. ***By default tomcat runs on port 8080 , we changed it to 8585 , easy to revert back to 8080 on changing in the properties file ,Automated hacking servers frequently scanning ports which are common , that is the reason we changed it to 8585.***

6. ***Also  Mosquitto mqtt server to  have SSL certificate  to safe guard to the topic messages and communications.***