Server Developer Test - **OwlMighty Games**

Overview & important notes:

The Goal of the home assignment is to test your abilities as a server developer & cloud expert, please take the time to complete the test to the best of your abilities.

In your assignment we will be looking at the following abilities:

- Code Quality
  - Code readability
  - Code structure
  - Errors & Edge cases
- Code Architecture
  - Provide solutions using OOP
  - Use of Inheritance and avoiding duplicate code
  - Creating code that can be easily scaled
- Finishing level
  - The assignment should have a finished look and feel
- Code performance
  - Code should be written in the most efficient way to support millions concurrent calls.
- X factor
  - Anything that could make your assignment stand out, Make it intuitive, fun, and attention grabbing!!

1.Simple Game Server

Prerequisites:

Your assignment is to create a simple game server on one of the following platforms : AWS, Google Cloud or Azure.

Overview:

We are going to create a server app that support the following functionality:

1. Authenticate Client Identity using OAuth 2.0
2. Register user properties and save them in a NOSQL Database
3. Retrieve saved user properties
4. Upload a user profile to S3
5. Retrieve user profile from S3

Implementation:

1. Authentication
    a. Create a POST endpoint <host>/login which will receive an email in its request body. The service will
        i. check if user exists, if not register a new user in the database
        ii. Will return an access token for future authentications
2. GET/POST user properties
    a. Create a POST endpoint <host>/registeruser that will receive a json in its request body with the following properties
        i. First name
        ii. Last name
        iii. Email
        iv. nickname
    b. The service should register the information in the database, if the information exists, the service should overwrite the information.
    c. Create a GET endpoint <host>/userinfo that will return a json in its response body with the properties mentioned above (first name, last name, etc…)
    d. The user identity should be determined by the access token provided by previous section
3. POST/GET user profile picture
    a. Create a POST endpoint <host>/uploadprofile that will receive an image in its request body.
        i. The service should upload the image to the storage provider, and store it under the identity of the user (according to the access token)

b. Create a GET endpoint <host>/getprofile that will return the profile picture in its response body

Please provide full code implementation, screenshots of cloud services configuration or written configuration (serverless framework) and full protocol for each section: how the request headers/body and response headers/body should look like.

2.Multiplayer Game Server Architecture

1. Expanding the simple game server from task 1, design a server architecture that supports *asynchronous tournaments. On top of the requirements in previous question, the server should:
    a. Support multi region distribution
    b. Will be able to scale and automatically balance peak request times
    c. The response time for the server should be extremely fast ~10-30ms
    d. Should be as cost efficient as possible
    e. Support Asynchronous tournaments

Please provide a detailed description on how you envision the server side architecture, together with diagrams.

*Asynchronous tournaments description:

1. X number of players can join a tournament at any given time.
2. Each player plays for a limited amount of time
3. When the time is up, the player is prompted to submit their score to the server
4. Once all players have joined the tournament and submitted their score, the player with the highest score will get a notification that they won the tournament.
5. There is no time limit for the tournament itself, meaning that players can join the tournament as long as the tournament is not full.
6. A tournament has predefined spots available, once all spots have been filled, it is considered to be a full tournament.