Project Title: "Multi-Cloud Serverless Collaborative Trivia Challenge Game"
Product Name: Trivia Titans
Duration: 60 working days (approx.)
Edited by: Ankush Mudgal | Bharat Shankaranarayanan | Rahul Macwan | Saurabh Dey
Date Given: May 15, 2023

Due Date: Multiple Deliverables (refer to Syllabus and Brightspace home page)

Tab 1: Important deliverables due date chart

Updates	Weekly (as directed by Lead TA)
Final Report	Aug 4, 2023 (11:59 pm)
Final Demo Video	Aug 4, 2023 (11:59 pm)
1:1 Q&A	During exam period

Team Size: 4 (min) - 5 (max)
Communication: Dedicated Teams Channel created by Lead TA

Definition: To develop a multi-cloud serverless online trivia game that allows users to form teams, compete against other teams in real-time, and track their progress on global and category-specific leaderboards. The game will provide a personalized experience by adapting to users' preferences and offering a diverse mix of questions tailored to their interests and skill levels. Additionally, administrators will have access to analytics tools for monitoring the game's performance and user engagement, enabling them to make data-driven decisions to improve the overall gaming experience.

Hypothetical Scenario:
John, Sarah, and three of their friends are all trivia enthusiasts who enjoy participating in quizzes together. They learn about a new online trivia game called "Trivia Titans," which allows users to form teams and compete against others in real-time. Intrigued, they decide to give it a try.

John signs up using his Google account, while Sarah opts to use her Facebook account. Their friends sign up using their email addresses. After completing the registration process, they create their user profiles and set their profile pictures. Since the system uses a 2-factor authentication, the second factor (three questions, and related answers) is recorded during the registration process.

John creates a new team called "The Quizzards" and invites Sarah and their friends to join. They all accept the invitations and become part of the team. Together, they explore the game lobby and find an upcoming trivia game focused on science and technology, one of their favorite categories. The game is scheduled to start in 10 minutes and has a 30-minute time limit.

As they wait for the game to begin, the team members chat using the in-app messaging feature, discussing their favorite science topics and sharing some interesting facts. Once the game starts, they are presented with a series of multiple-choice trivia questions, each with a 20-second time limit. The questions cover various subcategories within science and technology and have different difficulty levels.

The team members collaborate by discussing the questions in real-time and sharing their knowledge. They request hints occasionally, which help them answer some of the more challenging questions. Throughout the game, they can see their team's progress and score, along with the real-time scores of other competing teams.

After the 30 minutes are up, the game ends, and "The Quizzards" find out that they have secured second place. They are thrilled with their performance and are eager to participate in more trivia games to improve their ranking. As they continue playing, they unlock various achievements and see their names rise on the global and category-specific leaderboards.

As the players become more engaged with "Trivia Titans," they customize their game preferences, choosing their favorite categories and adjusting notification settings. The app personalizes their experience, providing a diverse and engaging mix of questions that cater to their interests and skill levels.

Meanwhile, the admins monitor the game's performance and user engagement, using the analytics tools provided by the Multi-Cloud Serverless Collaborative Trivia Challenge Game platform. They analyze the gameplay data, identify trends and patterns, and make data-driven decisions to improve the game experience for all users.

Front end (Non-essential module considering course objective)

Build a front-end application using suitable framework and use it to call backend services – Use of React as a front-end can be an option. You can use any other web technologies as well.
Hosting of entire front-end application or user facing interface should be done as a microservice on GCP Cloud Run.

Back end (Essential modules)

1. **User Authentication:**
    - Sign up and log in using social media accounts or email addresses
    - Password recovery and reset if provided by the cloud service
    - Register the second factor (3 pre-defined questions and answers) to the NoSQL database, and run a code to access and validate the question answers.

Required cloud services (Justification needed for the services used):

Option 1: [1st factor] AWS Cognito + [2nd factor] Firestore (for storing Q&A) + Cloud Function (Q&A validation)
Option 2: [1st factor] GCP - Firebase Authentication + [2nd factor] DynamoDB (for storing Q&A) + AWS Lambda (Q&A validation)

2. **User Profile Management:**
    - Edit personal information (e.g. profile picture, name, and contact information etc.)
    - View user statistics (e.g., games played, win/loss ratio, and total points earned)
    - View and manage team affiliations
    - View and compare achievements with other users

Required cloud services (Justification needed for the services used):

Option 1: GCP – Firestore + Cloud Functions
Option 2: AWS – DynamoDB + Lambda Functions

3. Team Management (Includes usage of AI):

- Create a team with AI (artificial intelligence) generated team name //Including a 3rd party using API
- Invite other users to join the team // sending invitation using Pub/Sub
- Accept or decline team invitations // sending acknowledgement using Pub/Sub
- View team statistics (e.g., games played, win/loss ratio, and total points earned)
- Manage team members (e.g., promote to admin, remove members, or leave the team)

Required cloud services (Justification needed for the services used):

Option 1: GCP – Firestore + Cloud Functions + Pub/Sub + ChatGPT (Open AI)
Option 2: AWS – DynamoDB + Lambda Functions + SQS + SNS + ChatGPT (Open AI)

4. Trivia Game Lobby:

- Browse and join available trivia games created by administrators
- Filter trivia games by category, difficulty level, or time frame
- View game details, such as the number of participants, time remaining, and a short description

Proposed cloud services (Justification needed for the services used):

AWS – DynamoDB || Lambda Functions || SQS || SNS || GCP – Firestore || Cloud Functions || Pub/Sub

5. In-Game Experience:

- Answer multiple-choice trivia questions within a specified time frame
- View real-time team scores
- Communicate with team members through a chat feature //instant messaging can be used
- Request and share hints with team members //instant messaging can be used
- View the correct answer and explanation after the time has elapsed for each question
- Track individual and team performance

Proposed cloud services (Justification needed for the services used):

AWS – DynamoDB || Lambda Functions || SQS || SNS || GCP – Firestore || Cloud Functions || Pub/Sub || AWS QuickSight || GCP DataStudio

6. Leaderboards:

- View global and category-specific leaderboards for teams and individual players
- Filter leaderboards by time frame (e.g., daily, weekly, monthly, or all-time)
- View detailed statistics for top-performing teams and players

Proposed cloud services (Justification needed for the services used):

AWS – DynamoDB || Lambda Functions || SQS || SNS || GCP – Firestore || Cloud Functions || Pub/Sub || AWS QuickSight || Looker Studio

7. Trivia Content Management (Administrators):

- Add, edit, and delete trivia questions, including category and difficulty level
- Create and manage trivia games with custom settings (e.g., categories, difficulty levels, and time frames)
- Monitor and analyze gameplay data and user engagement

Proposed cloud services (Justification needed for the services used):

AWS – DynamoDB || Lambda Functions || SQS || SNS || GCP – Firestore || Cloud Functions || Pub/Sub || AWS QuickSight || Looker Studio

8. Notifications and Alerts:

- Receive notifications for game invites, team updates, and new trivia game availability //This is not an additional task, it is also mentioned in other tasks.
- Get alerts for achievements unlocked and leaderboard rank changes

Proposed cloud services (Justification needed for the services used):

AWS – DynamoDB || Lambda Functions || SQS || SNS || GCP – Firestore || Cloud Functions || Pub/Sub

9. Automated Question Tagging:

- Automatically tag each trivia question with relevant categories based on its content, such as sports, computer science, general knowledge, or entertainment.

Required cloud services (Justification needed for the services used):

Option 1: AWS – Comprehend + Lambda + DynamoDB/ Firestore
Option 2: GCP - Natural Language API + Cloud Function + Firestore/ DynamoDb

10. Virtual Assistance

- Implement a chatbot to provide navigation support //To help users locate signup and game playing options
- Another Chatbot option performs a dynamic database search for scores based on entered Team name

Required cloud services (Justification needed for the services used):

AWS –Lambda + DynamoDB/ Firestore + AWS Lex