



School of Computing

Diploma in Information Technology

Assignment: CA2

Class: DIT/FT/3A/02

Lecturer: Mr Choo Jek Bao

By: Muhd Muqtasidin Bin Ahmad Eunus (1846809)

Muhd Husni Bin Mohd Haron (1828696)

Edwin Low En Lin (1851830)

Tay Sui Sheng (1845192)

Goh Yi Fan, Amos (1851942)

Github Repository Link: https://github.com/edee17/CSC_Assignment_2

Table Of Contents

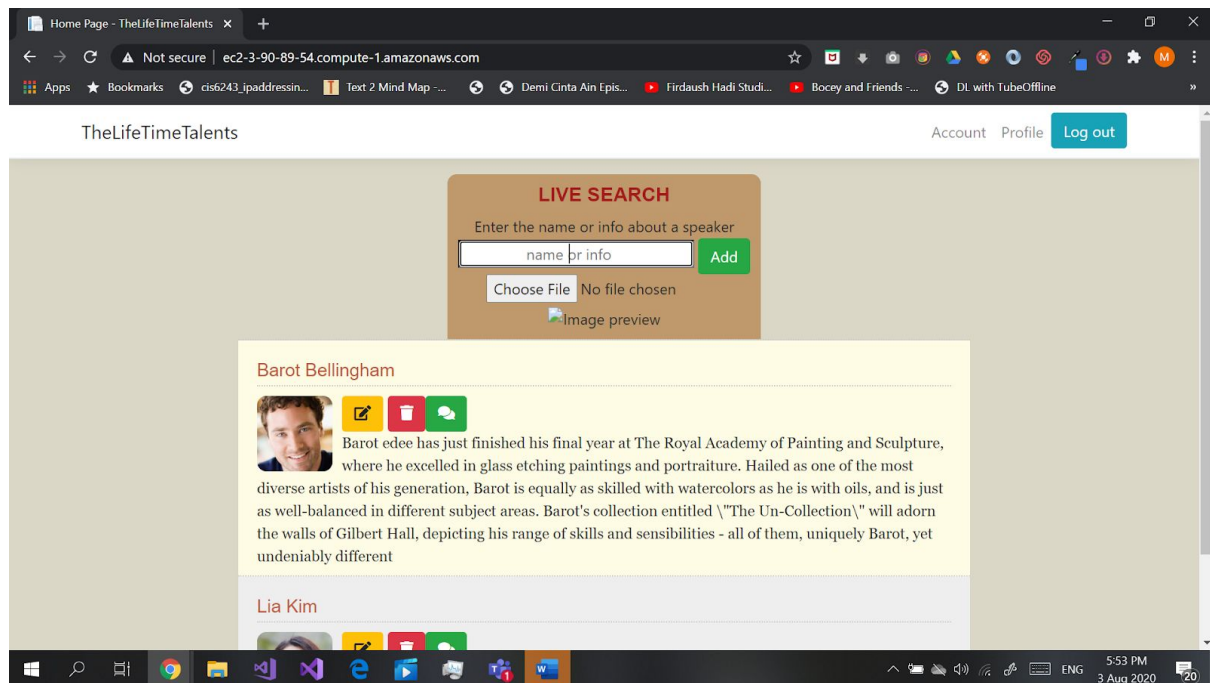
Virtual Server	4
Host Talents's SaaS web app	4
Screenshot	4
Payment Management	5
Accept daily subscription payment from users to differentiate between Free Users and Paid Users	5
API Documentation	5
Sequence Diagram	6
Discussion Management	7
Enable real time commenting for Paid users	7
Sequence Diagram	7
Storage & Image Recognition	8
Upload Talents images & Verify that the uploaded image is not NSFW	8
Postman Screenshots	8
Upload Talent Image to S3	8
S3 Bucket	10
SQL Database	11
Store Talents data such as image URL and talent details	11
API Documentation	11
Postman Screenshots	12
Get All Talents	12
Get Talent	12
Add Talent	13
Edit Talent	14
Delete Talent	15
NoSQL Database	16
Store Users such as subscription plan, last paid, and/or session data	16
API Documentation	16
Postman Screenshots	16
Data Lifecycle Management	17
Export manually Talents and Users data, import to data warehouse for future Business Intelligence (BI) use	17
Screenshots	17
Create Talent.csv in S3 bucket	17
Link S3 bucket to RedShift using AWS Glue	17
Export as CSV and put it in Redshift:	18

API Management	18
Publish/Deploy your Talents data REST API	18
API Documentation	18
Postman Screenshots	19
Cross Browser Testing (LambdaTest)	20
Show ability to use cloud cross browser testing tool to auto test web app and submit screenshots as proof	20
Screenshots	20
System Diagram	23
Draw a detailed architecture system diagram to illustrate your SaaS App	23

Virtual Server

Host Talents's SaaS web app

Screenshot



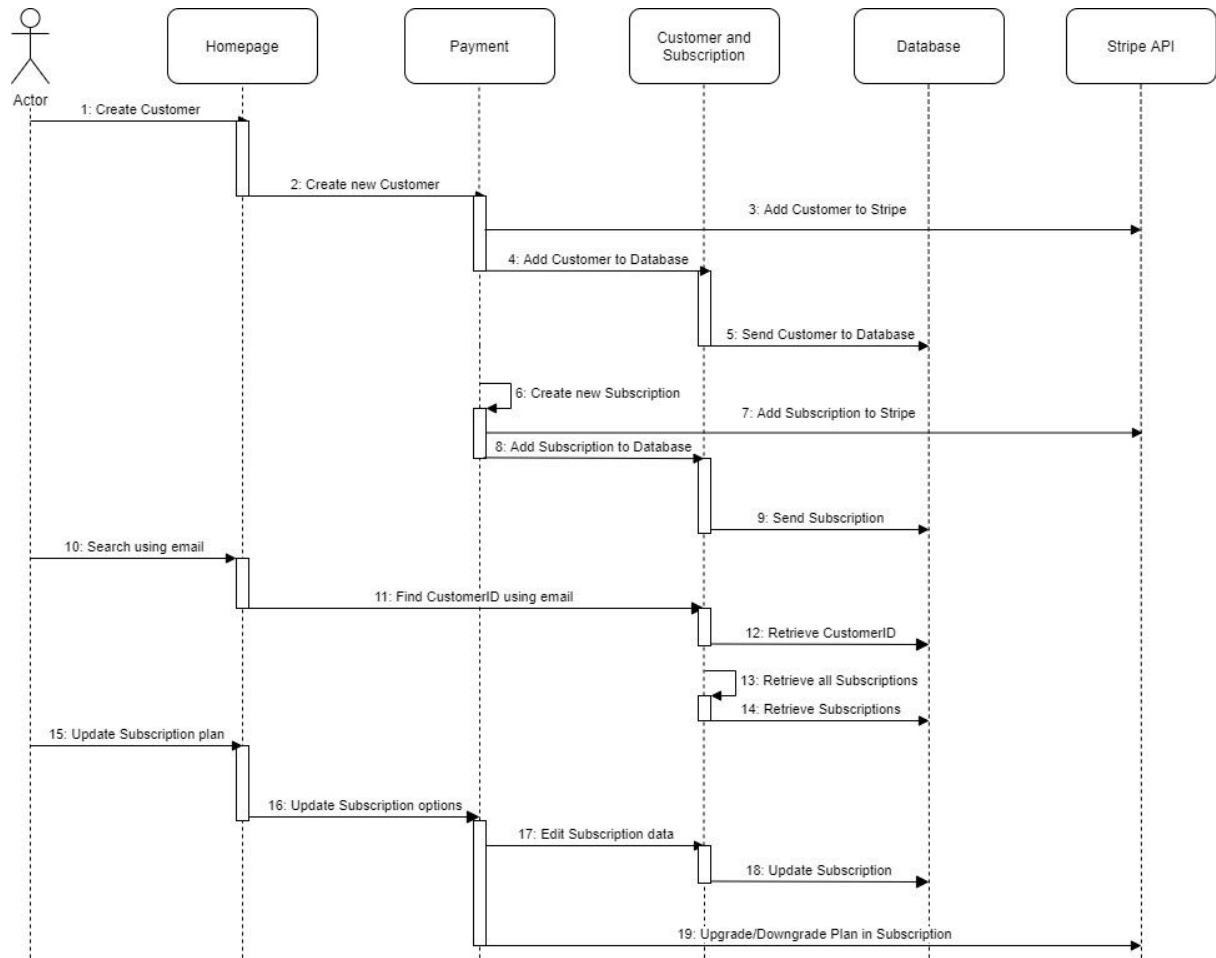
Payment Management

Accept daily subscription payment from users to differentiate between Free Users and Paid Users

API Documentation

1. PaymentController
 - a. Subscribe
 - i. Subscribe and create new customer to a product
 - ii. Method: Post
 - iii. Url: /Payment/Subscribe/
 - iv. Params:
 1. string cardEmail
 2. string plan
 3. string stripeToken
 - b. Upgrade
 - i. Upgrade or downgrade customer's subscription
 - ii. Method: Post
 - iii. Url: /Payment/Upgrade/
 - iv. Params:
 1. string subKey
 2. string plan

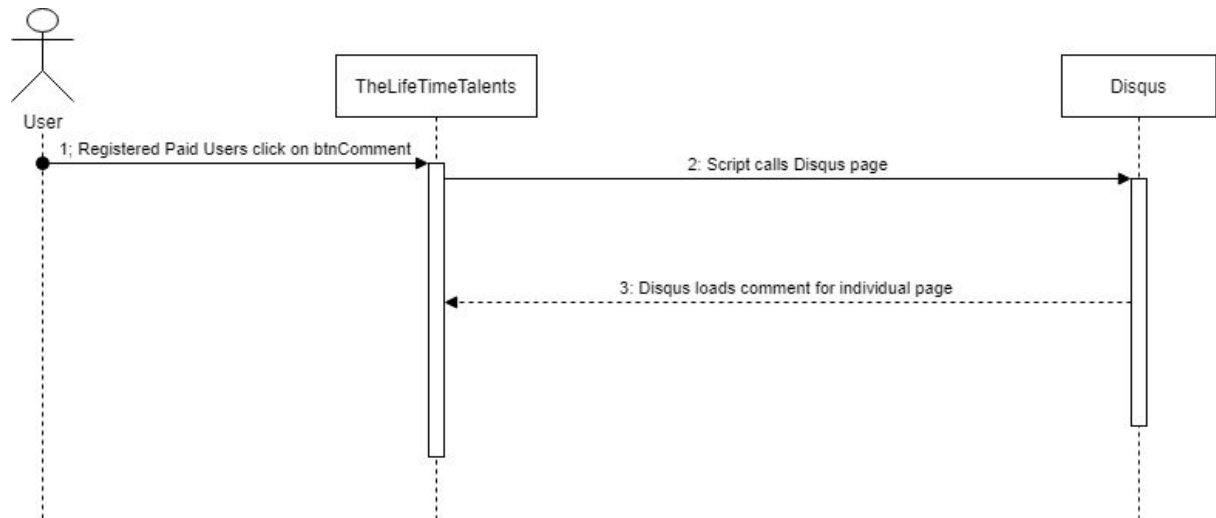
Sequence Diagram



Discussion Management

Enable real time commenting for Paid users

Sequence Diagram



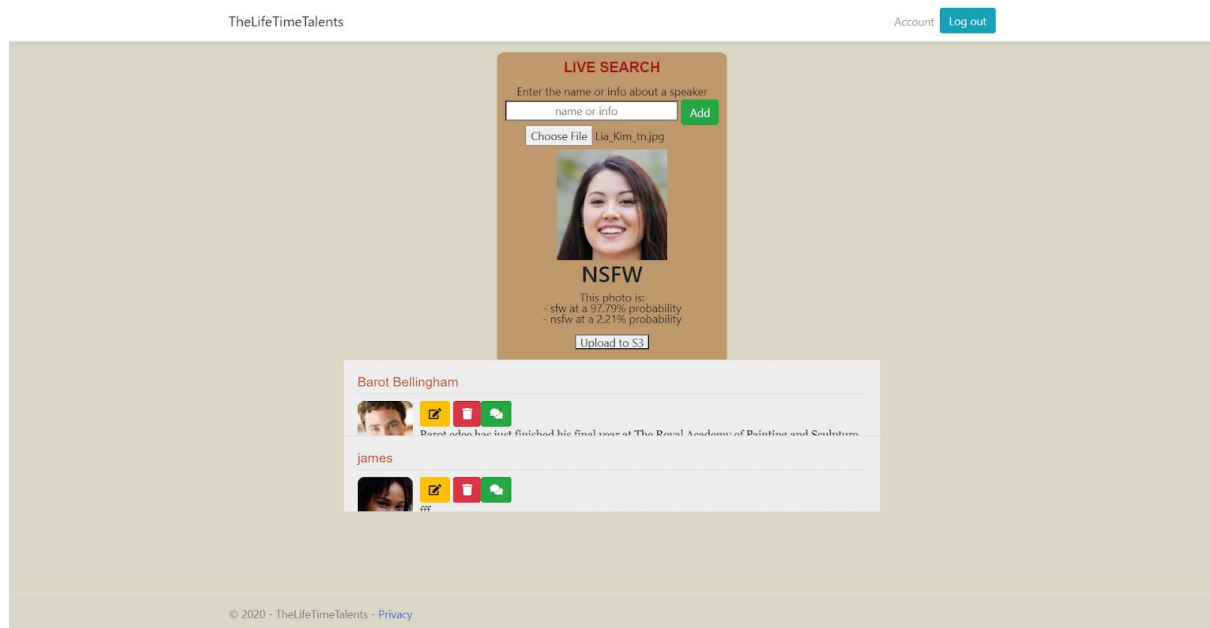
Storage & Image Recognition

Upload Talents images & Verify that the uploaded image is not NSFW

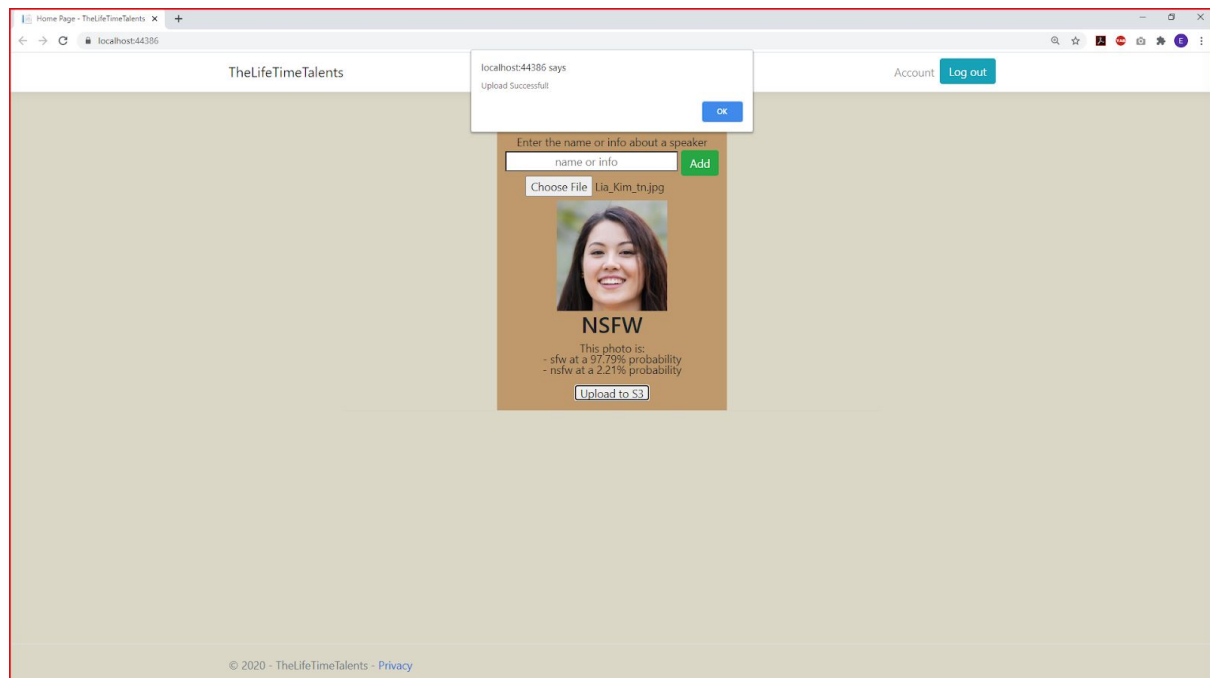
Postman Screenshots

Upload Talent Image to S3

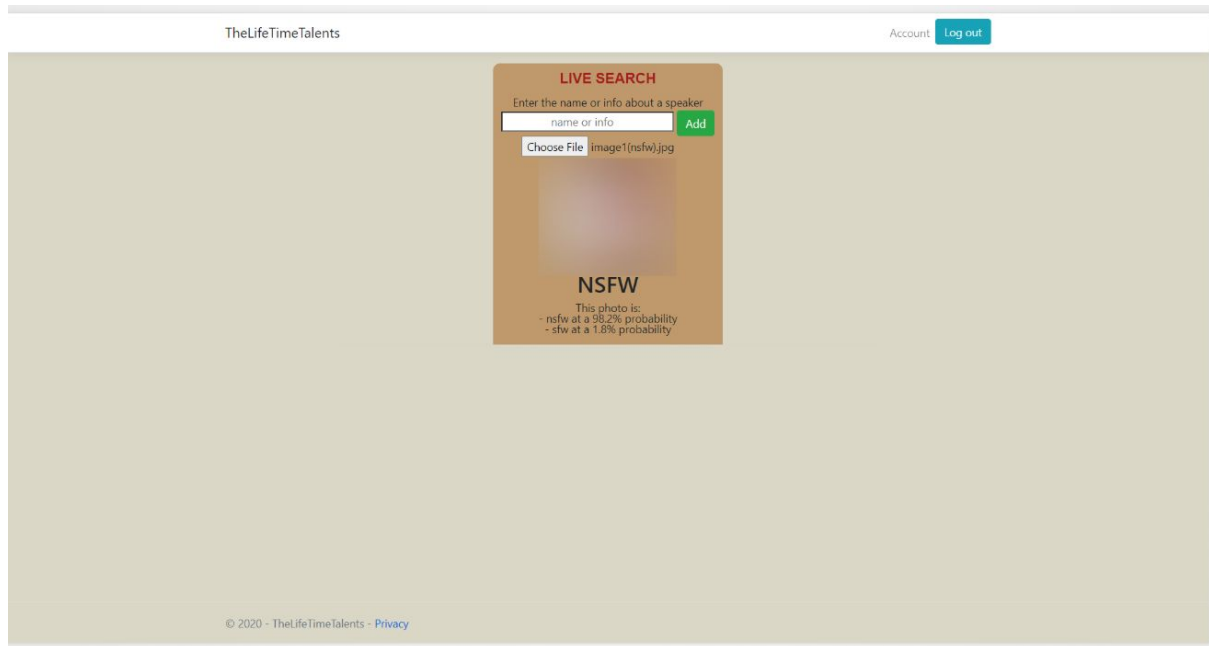
SFW:



Successful:



NSFW:



S3 Bucket

aws

Services ▾ Resource Groups ▾

🔔 vocstartsoft/user353764=muh... ▾ Global ▾ Support ▾

Amazon S3 > csctalents2

csctalents2

Overview Properties Permissions Management Access points

🔍 Type a prefix and press Enter to search. Press ESC to clear.

📁 Upload










➕ Create folder

⬇️ Download

⌵ Actions ▾

US East (N. Virginia) 🔁

Viewing 1 to 9

<input type="checkbox"/> Name ▾	Last modified ▾	Size ▾	Storage class ▾
<input type="checkbox"/>  2299786328467802896_4868071309.jpg	Jul 14, 2020 10:57:43 PM GMT+0800	145.9 KB	Standard
<input type="checkbox"/>  Barot_Bellingham_tn.jpg	Jul 9, 2020 4:00:48 PM GMT+0800	12.5 KB	Standard-IA
<input type="checkbox"/>  Ben_Daly_tn.jpg	Jul 9, 2020 4:20:22 PM GMT+0800	821.7 KB	Standard-IA
<input type="checkbox"/>  Hillary_Goldwynn_tn.jpg	Jul 14, 2020 12:39:26 PM GMT+0800	10.8 KB	Standard
<input type="checkbox"/>  Jonathan_Ferrar_tn.jpg	Jul 14, 2020 2:06:25 PM GMT+0800	13.0 KB	Standard
<input type="checkbox"/>  LaVonne-LaRue_tn.jpg	Jul 20, 2020 1:57:28 PM GMT+0800	12.2 KB	Standard
<input type="checkbox"/>  Lia_Kim_tn.jpg	Aug 3, 2020 2:18:55 PM GMT+0800	19.4 KB	Standard
<input type="checkbox"/>  Michael_Kim_tn.jpg	Jul 14, 2020 12:45:20 PM GMT+0800	854.3 KB	Standard
<input type="checkbox"/>  boy.jpg	Jul 20, 2020 2:03:37 PM GMT+0800	1015.3 KB	Standard

Viewing 1 to 9

Feedback

🌐 English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

SQL Database

Store Talents data such as image URL and talent details

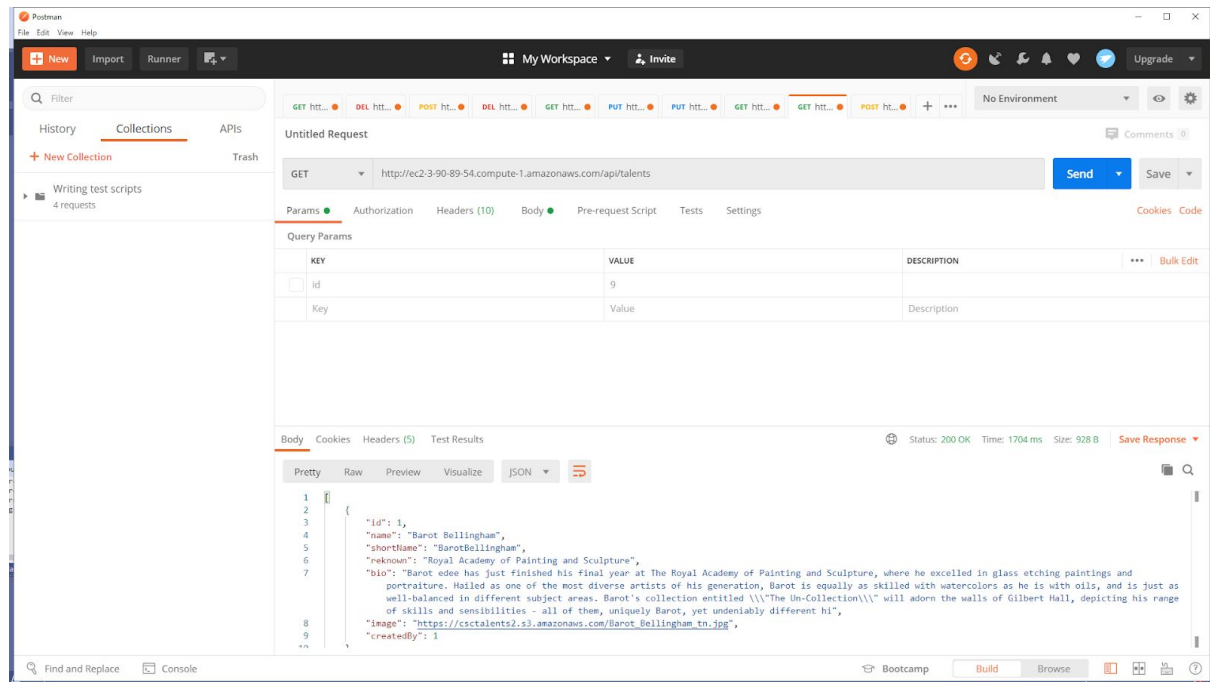
API Documentation

1. Get All Talents
 - a. Gets all talents
 - b. Method: GET
 - c. URL: api/talents
2. Get Talent
 - a. Get 1 talent specified by ID
 - b. Method: GET
 - c. URL: api/talents/{id}
 - d. Params
 - i. int id
3. Add Talent
 - a. Adds a talent
 - b. Method: POST
 - c. URL: api/talents/Add
 - d. Params
 - i. Talent talent
4. Edit Talent
 - a. Edits an existing talent in the repository
 - b. Method: PUT
 - c. URL: api/talents/{id}
 - d. Params
 - i. int id
 - ii. Talent talent
5. Delete Talent
 - a. Deletes a talent
 - b. Method: DELETE
 - c. URL: api/talents/{id}
 - d. Params
 - i. int id

Postman Screenshots

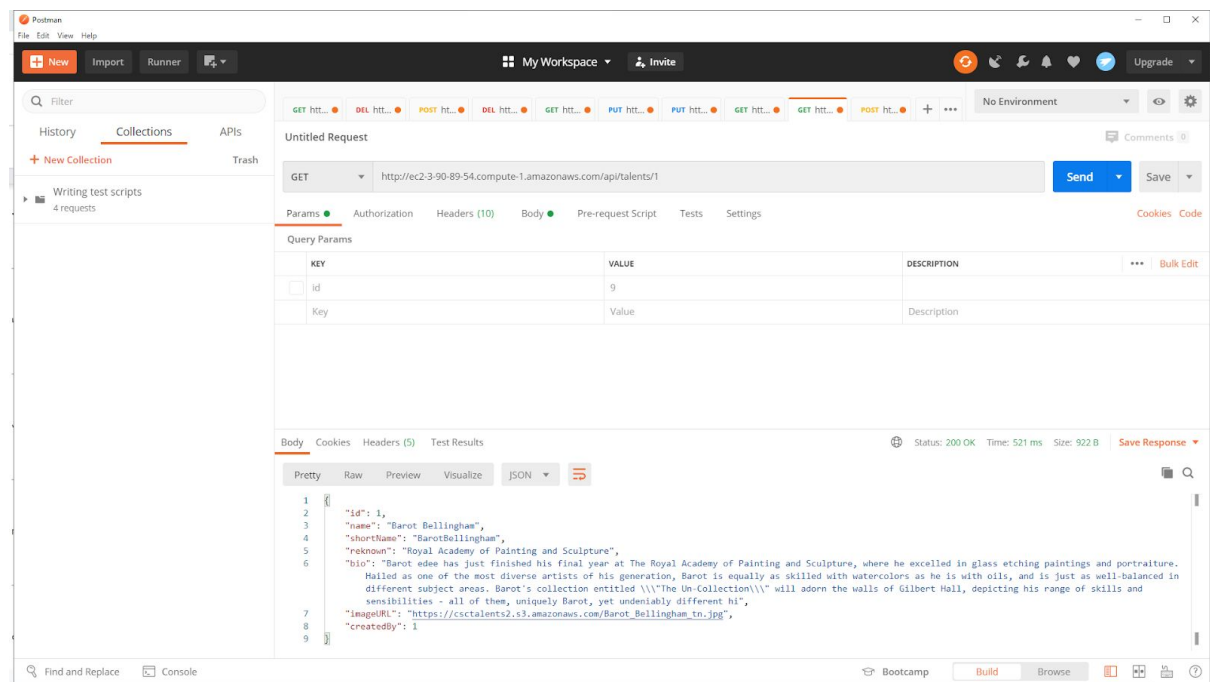
Get All Talents

Success:

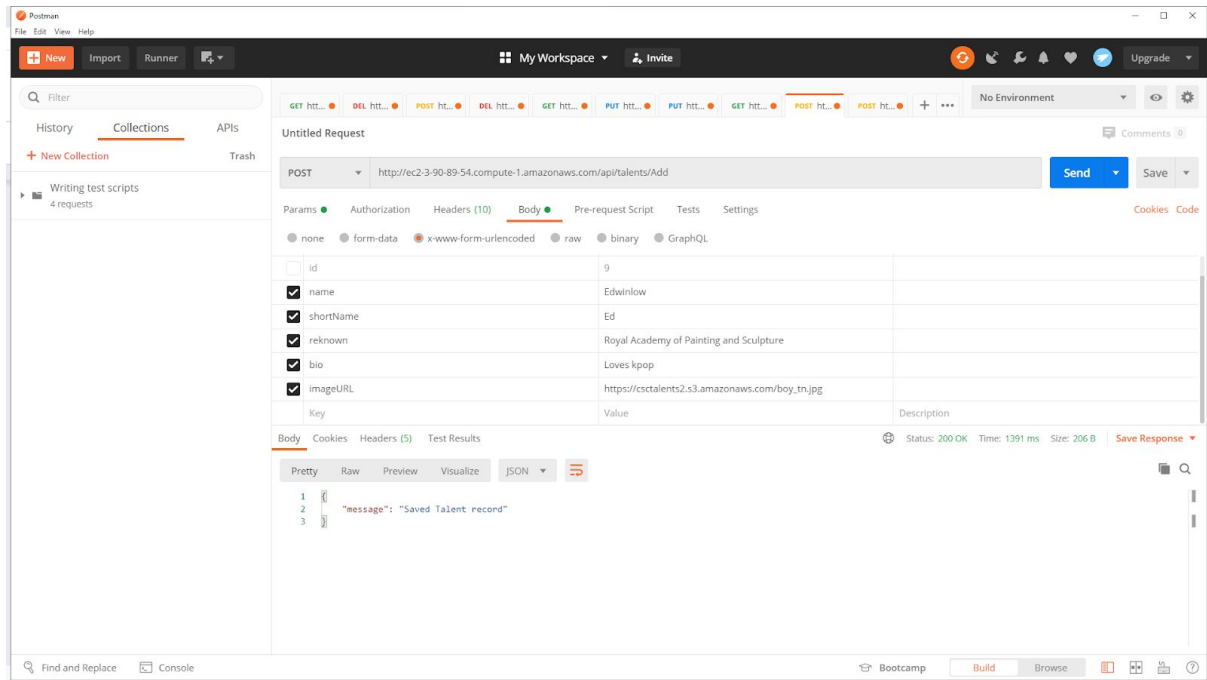


Get Talent

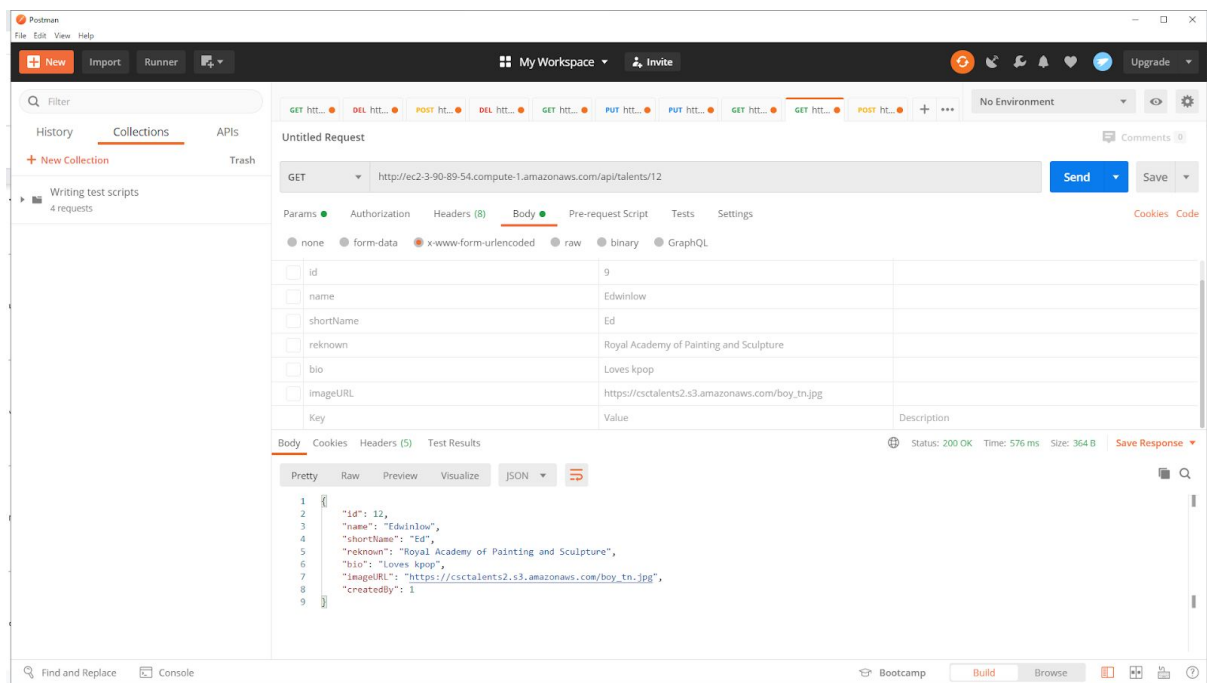
Success:



Add Talent Success:



Result



Edit Talent

The screenshot shows the Postman application with a PUT request configured to update a talent record. The URL is `http://ec2-3-90-89-54.compute-1.amazonaws.com/api/talents/1`. The request body is in JSON format, containing the following data:

Key	Value
id	1
name	Edwinlow
shortName	Edwin
reknown	Royal Academy of Painting and Sculpture boy
bio	Loves dancing
imageURL	<code>https://csctalents2.s3.amazonaws.com/boy.jpg</code>

The response is displayed in the bottom panel, showing a JSON message: `"message": "Edited talent record of Edwinlow"`. The status is 200 OK, with a response time of 293 ms and a size of 219 B.

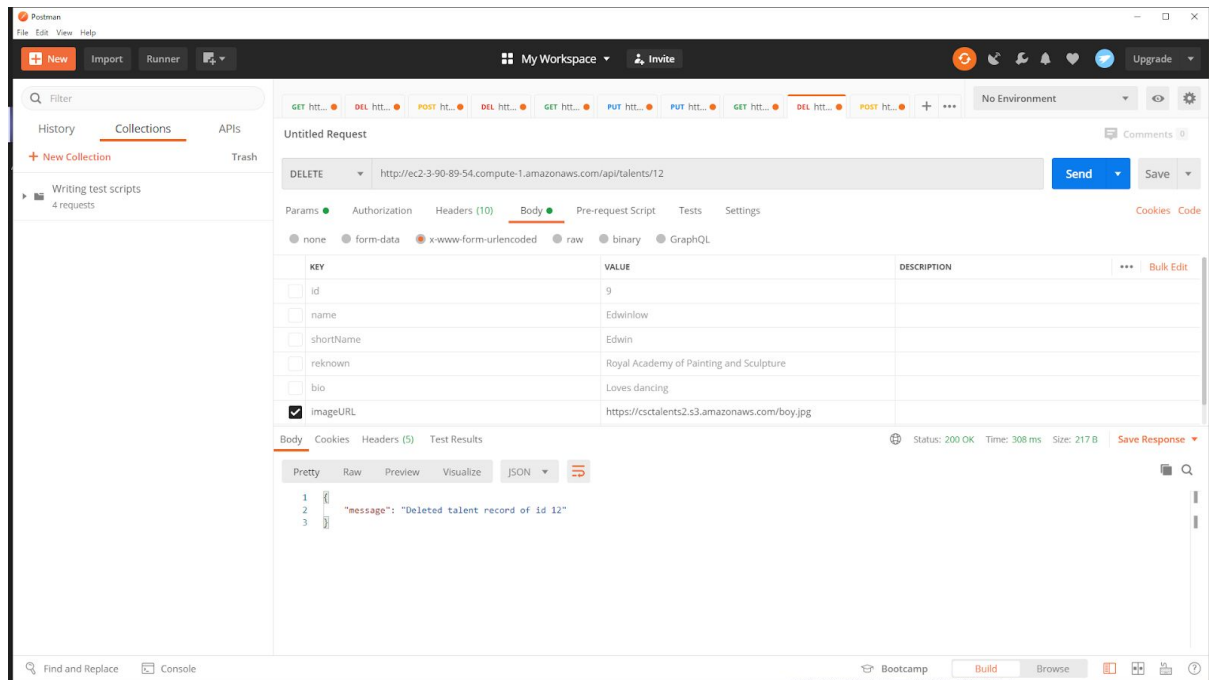
Result

The screenshot shows the Postman application with a GET request configured to retrieve a talent record. The URL is `http://ec2-3-90-89-54.compute-1.amazonaws.com/api/talents/1`. The response is displayed in the bottom panel, showing a JSON object with the following data:

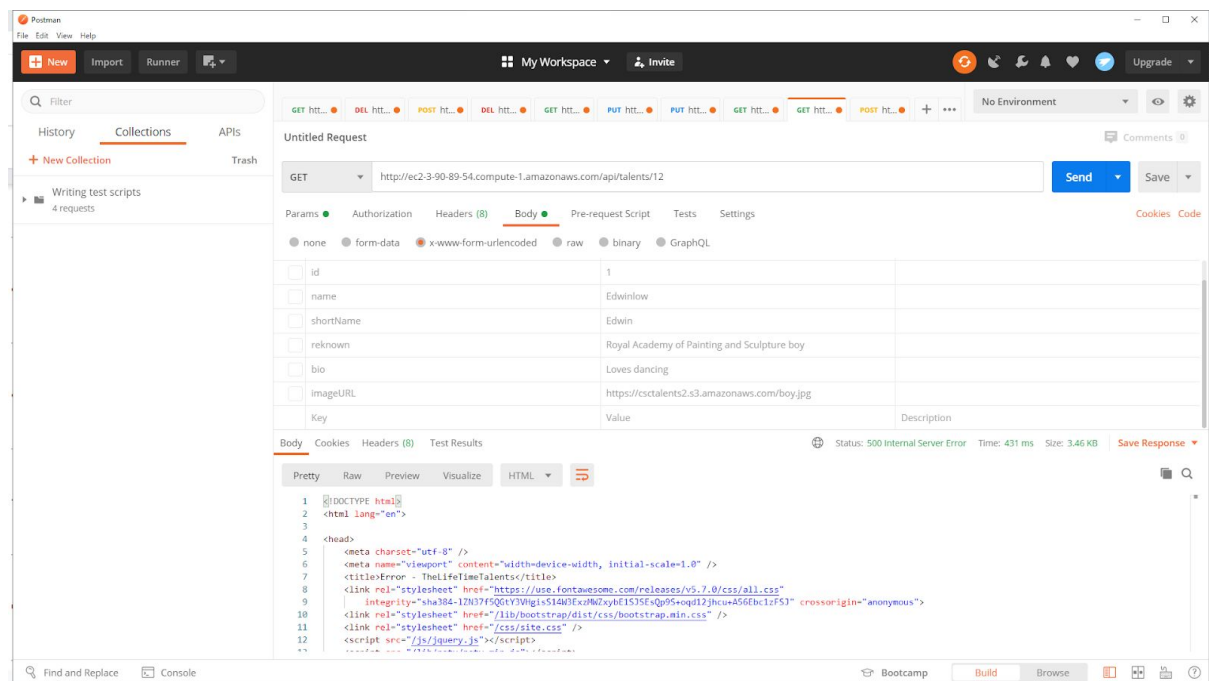
```
1 {
2   "id": 1,
3   "name": "Edwinlow",
4   "shortName": "Edwin",
5   "reknown": "Royal Academy of Painting and Sculpture boy",
6   "bio": "Loves dancing",
7   "imageURL": "https://csctalents2.s3.amazonaws.com/boy.jpg",
8   "createdBy": 1
9 }
```

The status is 200 OK, with a response time of 285 ms and a size of 370 B.

Delete Talent Success:



Result:



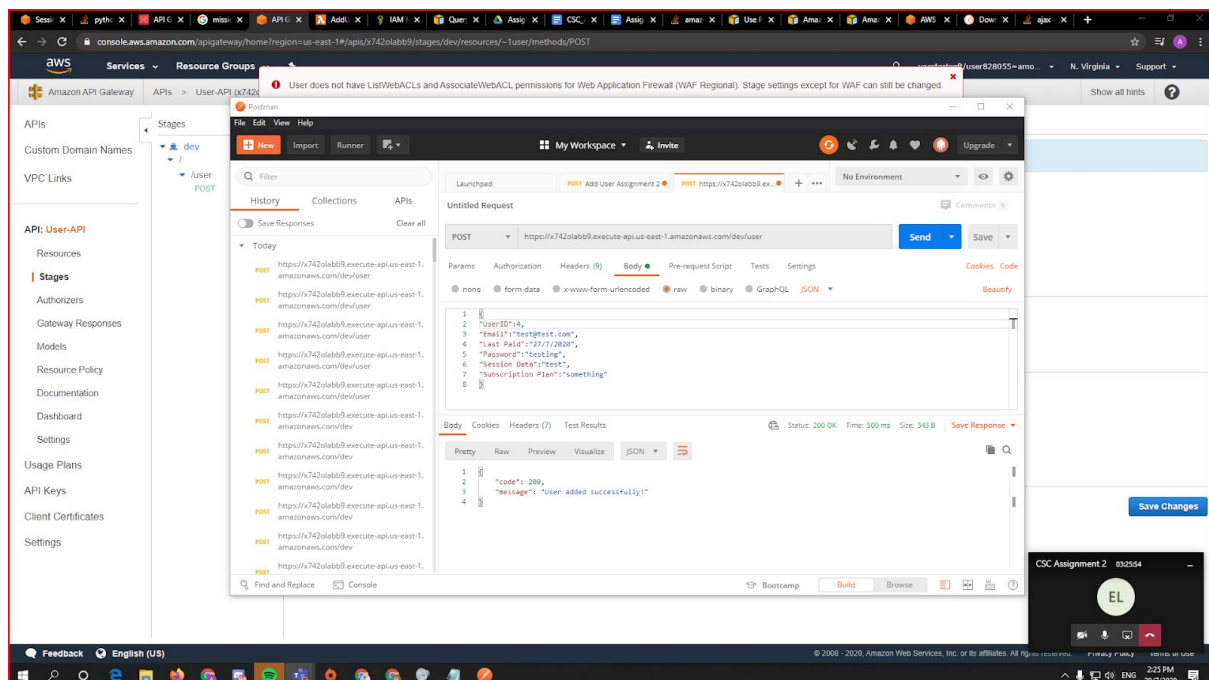
NoSQL Database

Store Users such as subscription plan, last paid, and/or session data

API Documentation

1. Add User
 - a. Adds a user to the database.
 - b. Method: POST
 - c. URL: /dev/user
 - d. Params
 - i. "UserID"
 - ii. "Email"
 - iii. "Last Paid"
 - iv. "Password"
 - v. "Session Data"
 - vi. "Subscription Plan"

Postman Screenshots

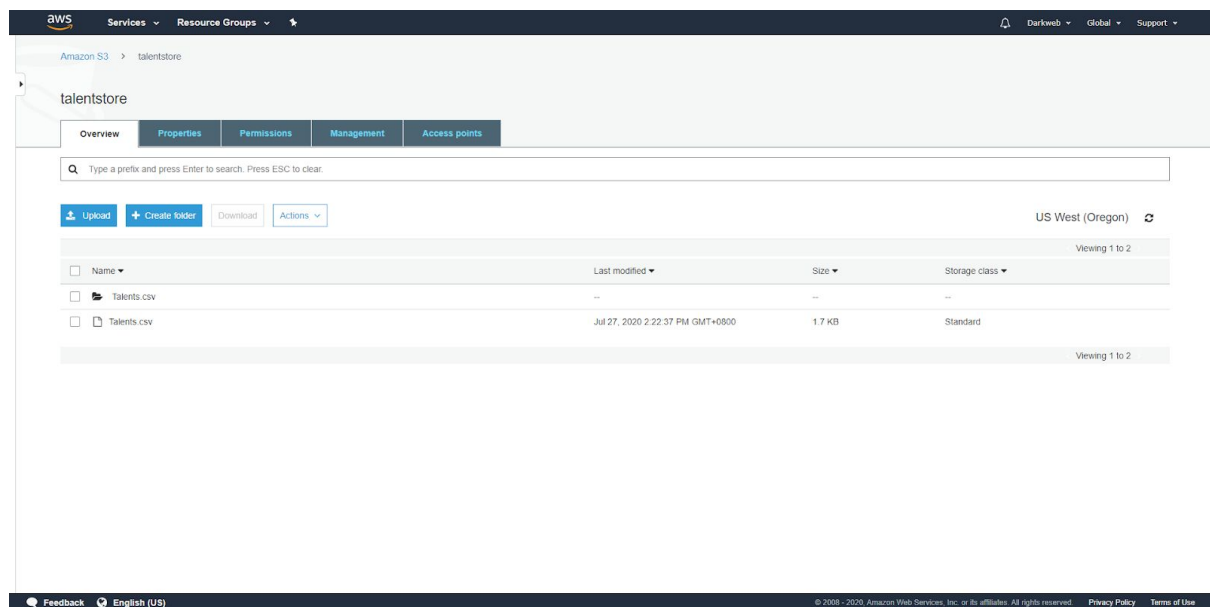


Data Lifecycle Management

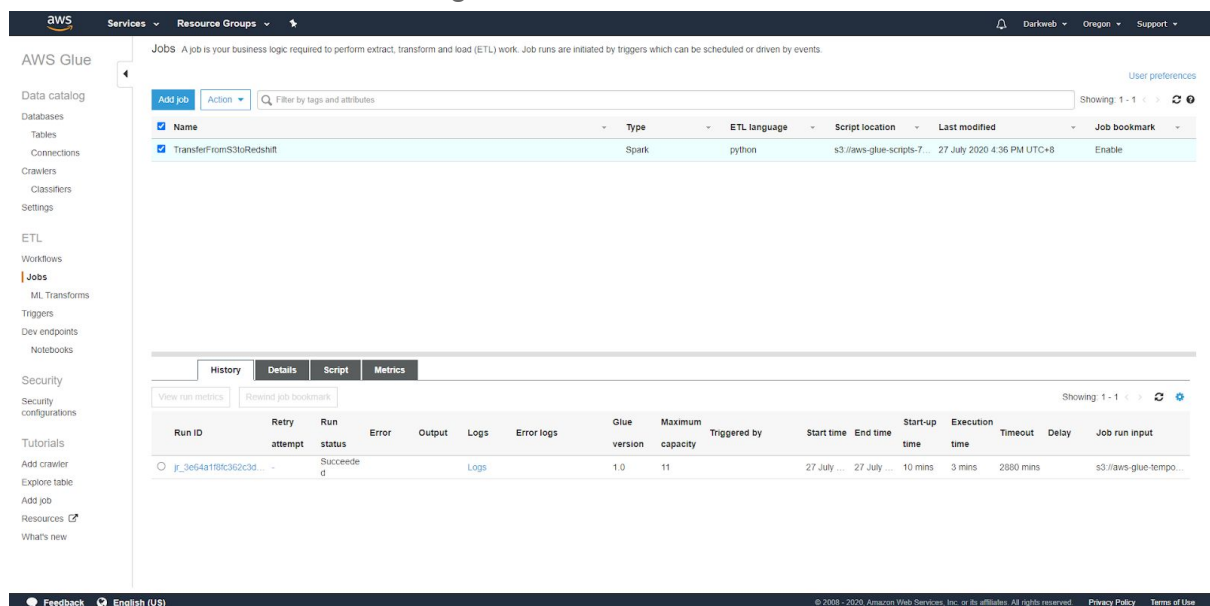
Export **manually** Talents and Users data, import to data warehouse for future Business Intelligence (BI) use

Screenshots

Create Talent.csv in S3 bucket



Link S3 bucket to RedShift using AWS Glue



Export as CSV and put it in Redshift:

The screenshot shows the AWS Data console interface. On the left, the 'Data objects' sidebar lists various services like Databases, Clusters, Queries, and more. The main area is the 'Query editor' for a Redshift cluster named 'talent'. It shows a table named 'talents_csv' with columns: id, name, shortname, reknown, bio, and imageurl. Below the table, there are tabs for 'Query history', 'Query results', and 'Table details'. The 'Table details' tab is active, showing the table's schema and a preview of its data. The data preview shows three rows of artist information.

id	name	shortname	reknown	bio	imageurl
1	Barot Bellingham	BarotBellingham	Royal Academy of Painting and Sculpture	Barot edee has just finished his final year at The Royal Academy of Painting and Sculpture, where he excelled in glass etching paintings and portraiture. Hailed as one of the most diverse artists of his generation, Barot is equally as skilled with watercolors as he is with oils, and is just as well-balanced in different subject areas. Barot's collection entitled \"The Un-Collection\" will adorn the walls of Gilbert Hall, depicting his range of skills and sensibilities - all of them, uniquely Barot, yet undeniably different	https://csctalents2.s3.amazonaws.com/Barot_Bellingham_tn.jpg
6	Edwin	Edwr	Student of SP on first sem	I love SP. School is good and also intern \"The Artist to Watch in 2012 by the London Review. Johnathan has already sold one of the highest priced commissions paid to an art student, ever on record. The piece, entitled Gratitude Resort, a work in oil and mixed media, was sold for \$750,000 and Jonathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations	https://csctalents2.s3.amazonaws.com/LaVonnie_LaRue_tn.jpg
2	Jonathan G. Ferrar II	Jonathan_Ferrar	Artist to Watch in 2012	The Artist to Watch in 2012 by the London Review, Johnathan has already sold one of the highest priced commissions paid to an art student, ever on record. The piece, entitled Gratitude Resort, a work in oil and mixed media, was sold for \$750,000 and Jonathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations	https://csctalents2.s3.amazonaws.com/Jonathan_Ferrar_tn.jpg

API Management

Publish/Deploy your Talents data REST API

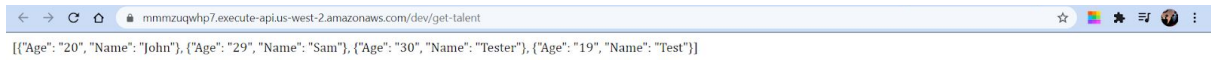
API Documentation

1. GET API
 - a. Get all data.
 - b. Method: Get
 - c. Url: <https://mmmzuqwhp7.execute-api.us-west-2.amazonaws.com/dev/get-talent>
2. POST API
 - a. Add new data.
 - b. Method: Post

- c. Url: <https://mmmzuqwhp7.execute-api.us-west-2.amazonaws.com/dev/add-talent>

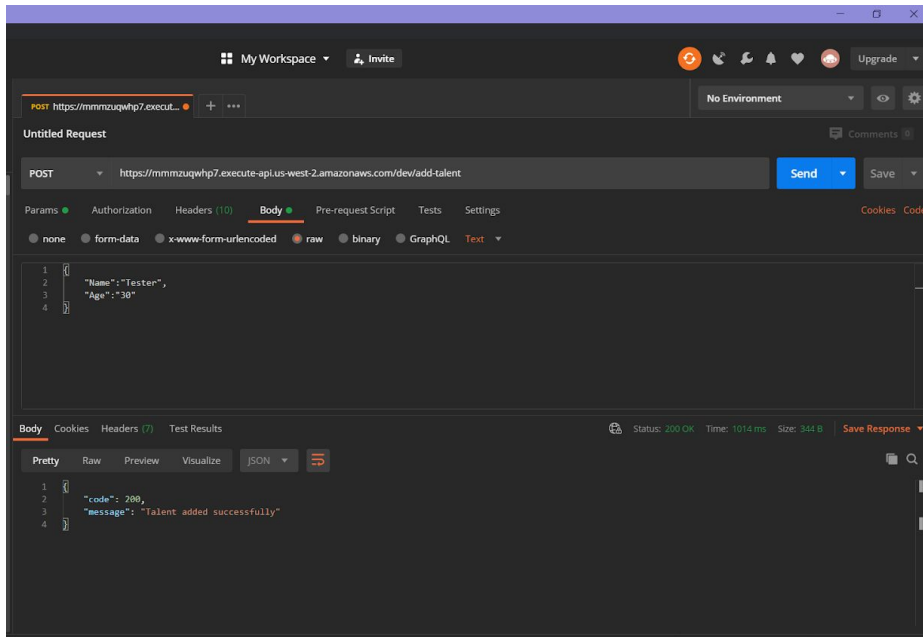
Postman Screenshots

[GET]



A screenshot of a web browser window. The address bar shows the URL `mmmzuqwhp7.execute-api.us-west-2.amazonaws.com/dev/get-talent`. The page content displays a JSON array of objects: `[{"Age": "20", "Name": "John"}, {"Age": "29", "Name": "Sam"}, {"Age": "30", "Name": "Tester"}, {"Age": "19", "Name": "Test"}]`.

[POST]

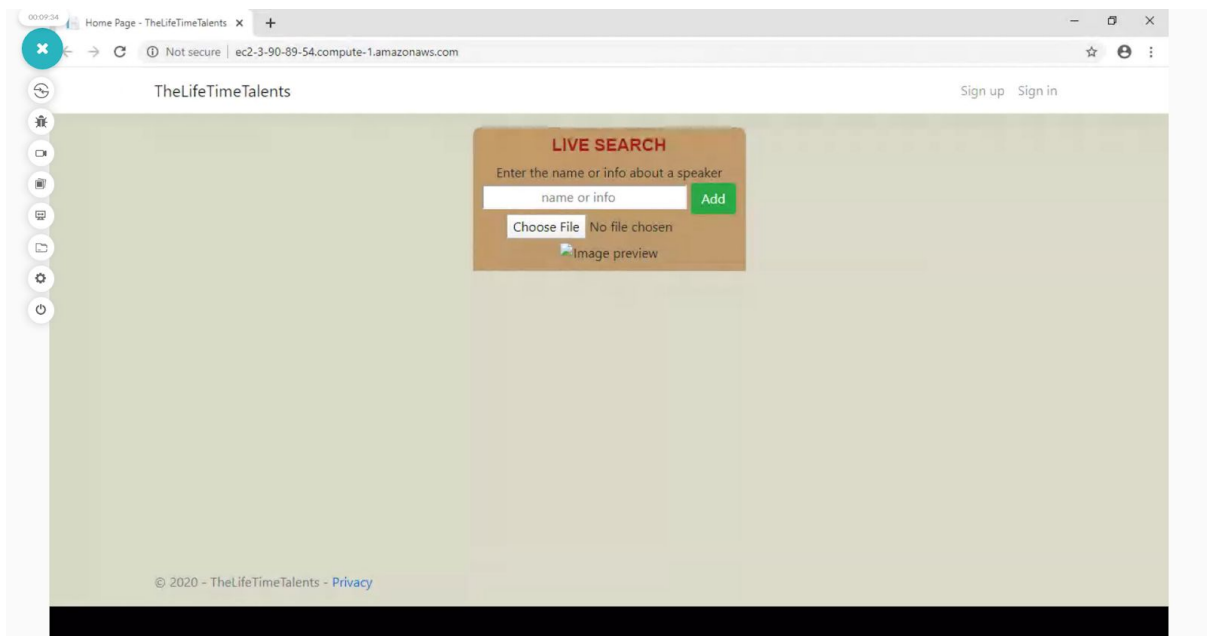


Cross Browser Testing (LambdaTest)

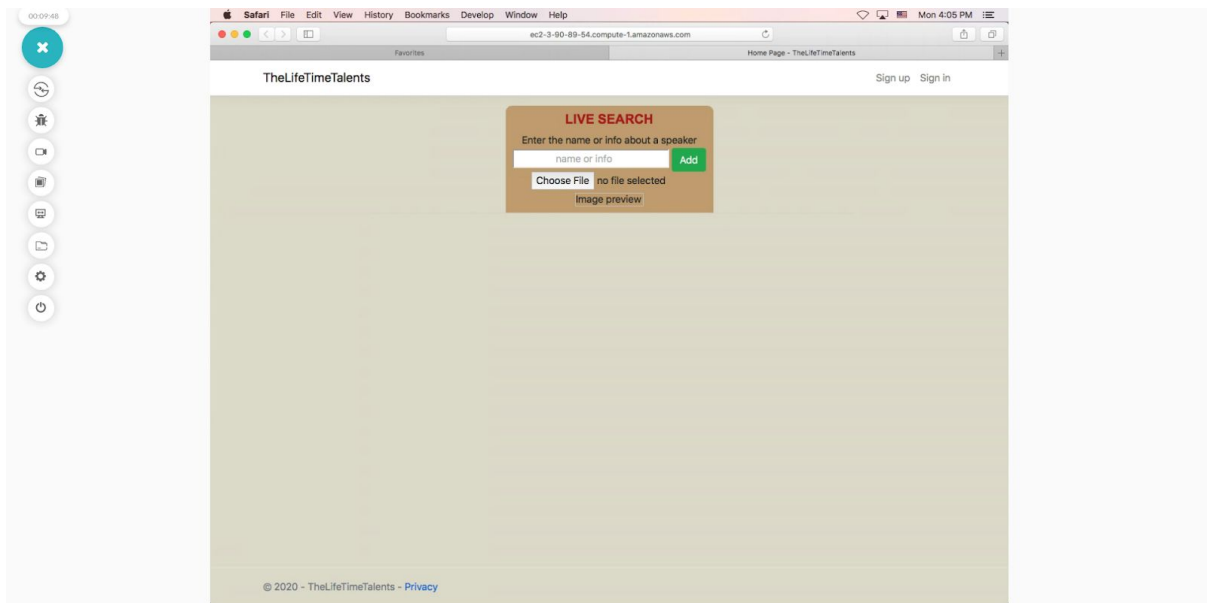
Show ability to use cloud cross browser testing tool to auto test web app and submit screenshots as proof

Screenshots

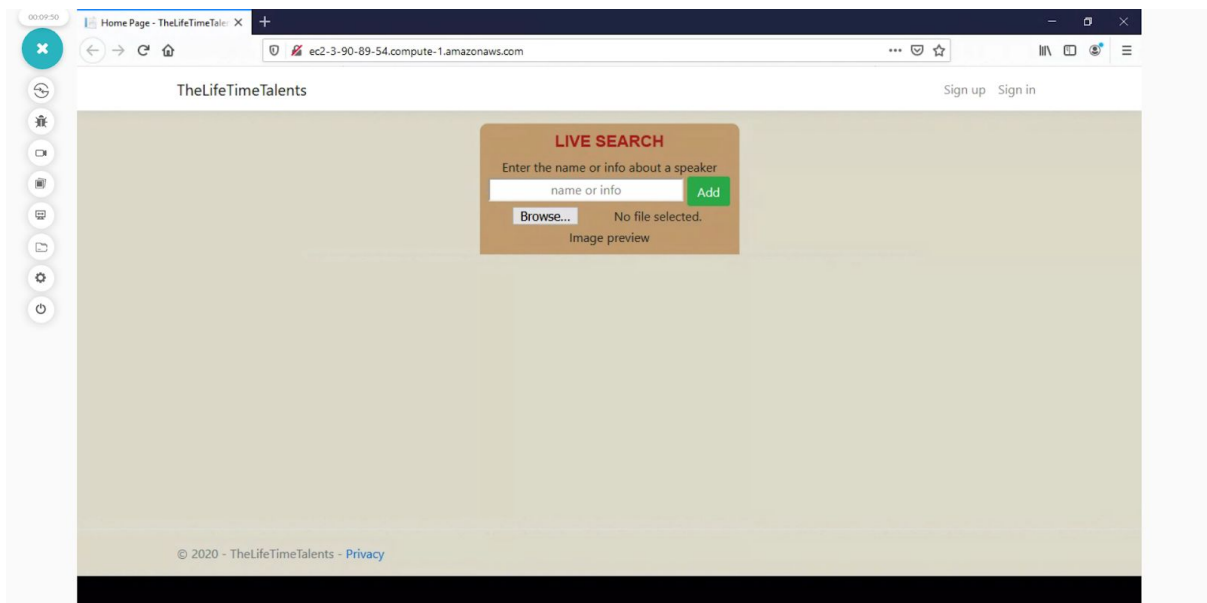
Chrome:



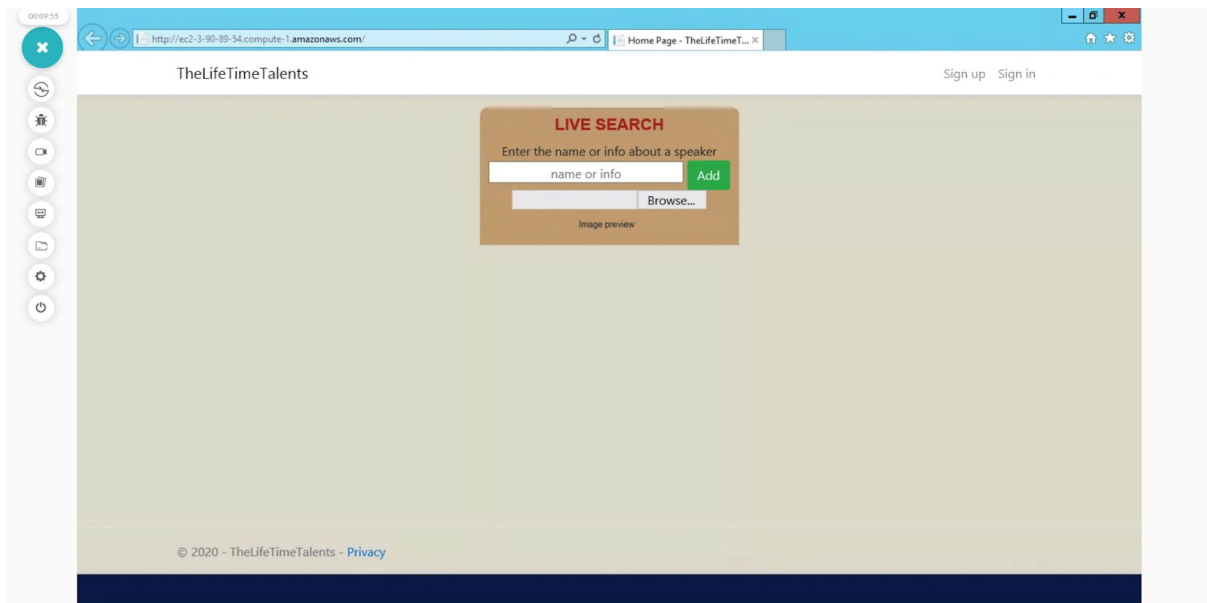
Safari:



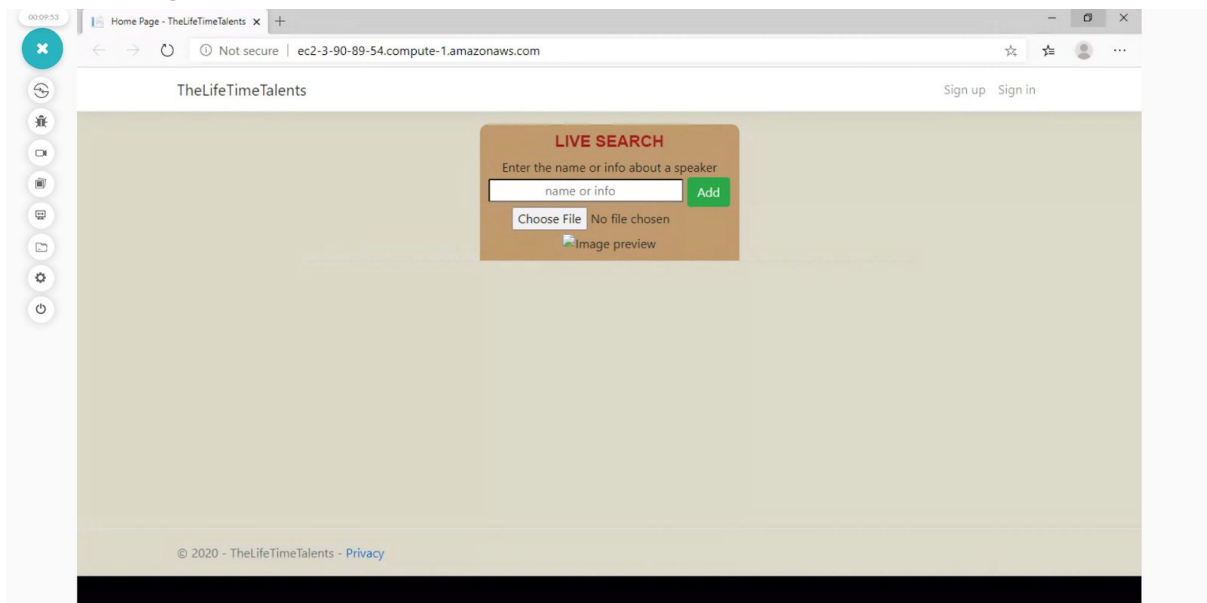
Firefox:



Internet Explorer:



Microsoft Edge:



System Diagram

Draw a detailed architecture system diagram to illustrate your SaaS App

