



School of Computing

Diploma in Information Technology

Assignment: CA1

Class: DIT/FT/3A/02

Lecturer: Mr Choo Jek Bao

By: Muhd Muqtasidin Bin Ahmad Eunos (1846809)

Muhd Husni Bin Mohd Haron (1828696)

Github Repository Link: https://github.com/edee17/CSC_Assignment_1-v2-

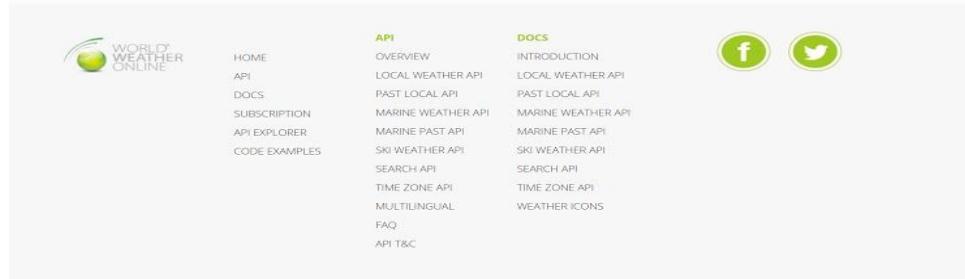
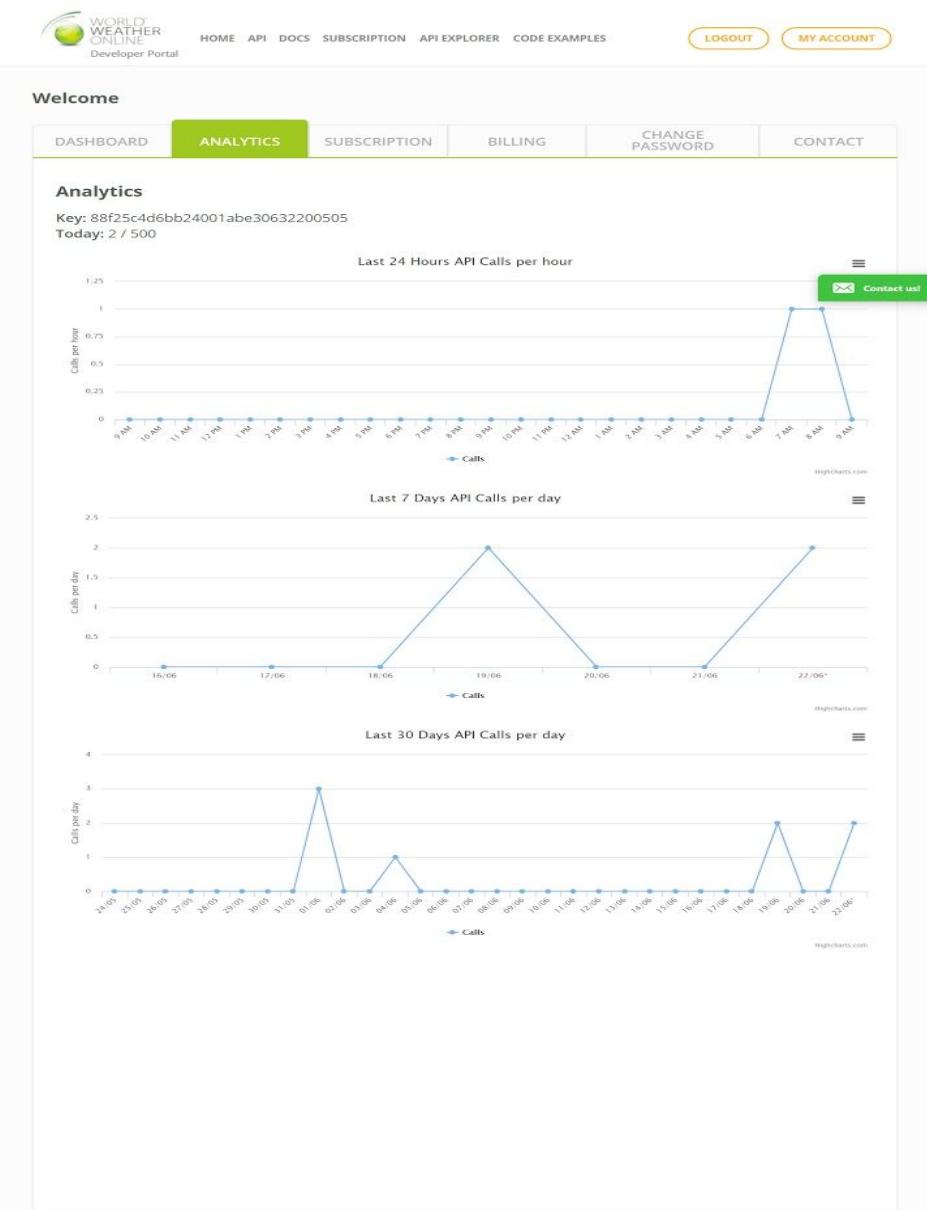
Table Of Contents

Task 1	3
Practical 1 Part D	3
Diagram	4
Task 2	5
Practical 2 Part 2	5
API Documentation	5
Postman Screenshots	6
Get Version	6
Get Multiple Names	7
Get All Products	8
Get Product	8
Get Products by Category	10
Add Product	10
Edit Product	11
Delete Product	11
Practical 3 Part 1	13
API Documentation	13
Postman Screenshots	13
Underposting	13
Overposting	14
Task 3	15
Web API 2 Security Authentication and Authorization using Bearer Token	15
API Documentation	15
Postman Screenshots	16
Get Values	16
Register	16
Login	18
Logout	19
Diagram	20
Task 4	21
Talents Search Functions & Secured Talent Search Web API	21
API Documentation	21
Postman Screenshots	22
Get All Talents	22
Get Talent	22
Add Talent	23
Edit Talent	24
Delete Talent	25

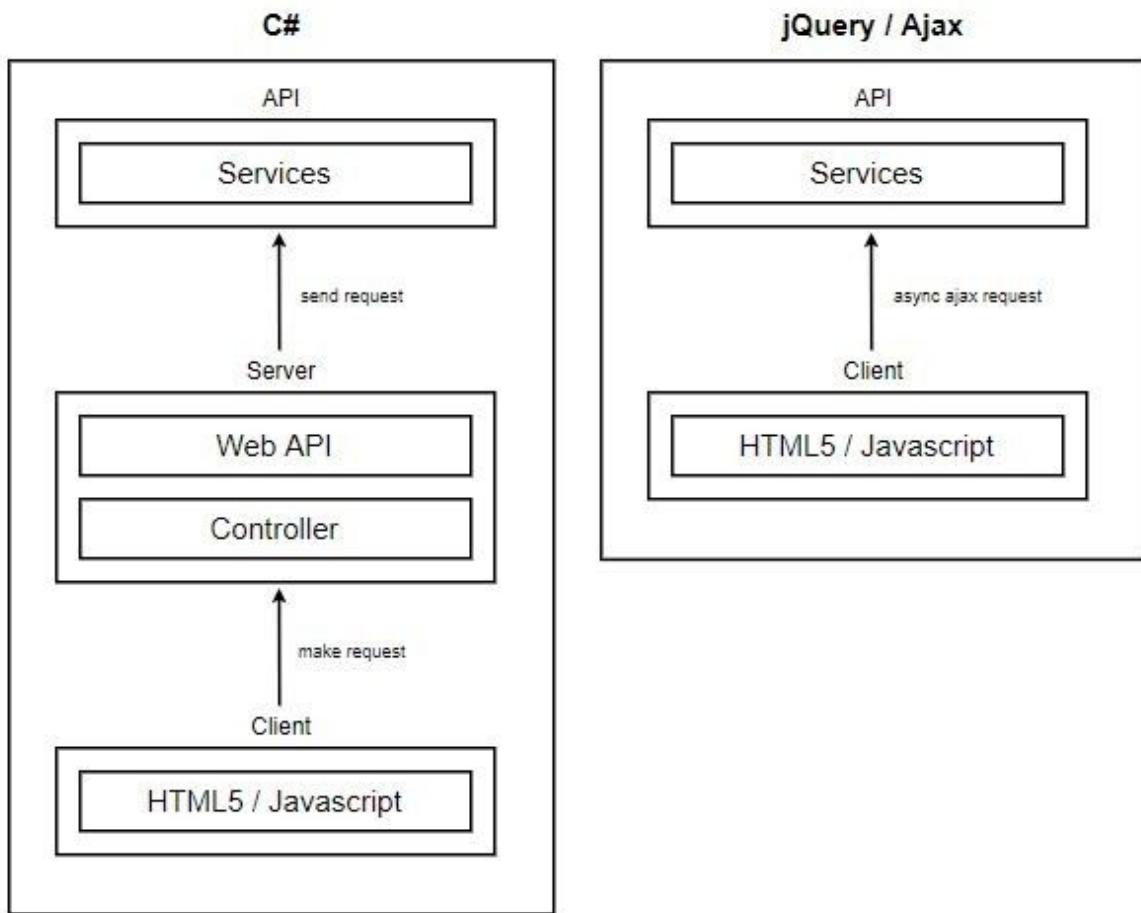
Diagram	25
Task 5	27
Store Talents Photo in AWS S3 & Generate bit.ly API	27
API Documentation	27
Postman Screenshots	27
Get list of Talents using HTTP	27
Get list of Talents using HTTPS	28
Store Talents in AWS S3 Bucket	30
Generate bit.ly API	31
Diagram	31
Task 6	33
Develop a web client that accept subscription payment using Stripe billing subscription API & Use Stripe webhooks to update your database when the subscription plan changes	33
API Documentation	33
Screenshots	34
Diagram	39
Task 7	40
Eventbrite integrates with your private Telegram Bots API	41
Screenshots	41
Diagram	47
Task 8	48
Develop a web client that allows user to upload receipts images and tag the receipts image by calling web service from Clarifai API	48
Screenshots	48
Diagram	49
Extra Feature	50
Dialogflow API (Text-To-Speech)	50
Screenshots	50

Task 1

Practical 1 Part D



Diagram



Task 2

Practical 2 Part 2

API Documentation

1. ProductsV1Controller
 - a. Get Version
 - i. Return the current version of API
 - ii. Method: GET
 - iii. URL: api/v1/products/version
 - b. Get Multiple Names
 - i. Returns the 3 Names written in query
 - ii. Method: GET
 - iii. URL:
api/v1/products/message?name1=Seb&name2=Max&name3=Charles
 - iv. Params:
 1. string message1
 2. string message2
 3. string message3
 - c. Get All Products
 - i. Return all the products
 - ii. Method: GET
 - iii. URL: api/v1/products
 - d. Get Product
 - i. Returns one product
 - ii. Method: GET
 - iii. URL: api/v1/products/{id}
 - iv. Params:
 1. Int id
 - a. Min value: 2
2. ProductsV2Controller
 - a. Get Products by Category
 - i. Returns all the products by category
 - ii. Method: GET
 - iii. URL: api/v2/products?category=[category]
 - iv. Params:
 1. string category
 - b. Add Product
 - i. Adds a product

- ii. Method: POST
- iii. URL: api/v2/products
- iv. Params:
 - 1. Product product
- c. Edit Product
 - i. Updates an existing product
 - ii. Method: PUT
 - iii. URL: api/v2/products/{id}
 - iv. Params:
 - 1. Int ID
- d. Delete Product
 - i. Deletes a product
 - ii. Method: DELETE
 - iii. URL: api/v2/products/{id}
 - iv. Params:
 - 1. Int ID

Postman Screenshots

Get Version

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'File', 'Edit', 'View', and 'Help' are visible. On the right, there are icons for 'Invite', 'Upgrade', and settings. The main workspace is titled 'Untitled Request'. A 'GET' request is selected, with the URL 'http://localhost:61235/api/v1/products/version' entered. The 'Params' tab is active, showing 'No Auth' under 'TYPE'. Below the request details, the response status is 'Status: 200 OK', time '102 ms', and size '457 B'. The response body is displayed in JSON format:

```

1 "hello",
2 "version 2",
3 "2"
4
5
  
```

The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Edge, and Teams.

Get Multiple Names

Success:

The screenshot shows the Postman application interface. In the center, there's a "Untitled Request" panel with a GET method selected. The URL is set to `http://localhost:61235/api/v1/products/message?name1=Seb&name2=Max&name3=Charles`. Below the URL, under the "Authorization" tab, it says "No Auth". The "Body" tab is selected, showing the response body as:

```
1 <html>
2 <body>Hello World name1 =Seb name2= Max name3=Charles is provided in path parameter</body>
4 </html>
```

At the bottom of the Postman window, the status bar shows "Status: 200 OK Time: 285 ms Size: 515 B". The taskbar at the bottom of the screen shows various icons, including a file explorer, browser, and system tray.

Missing Parameters:

The screenshot shows the Postman application interface. In the center, there's a "Untitled Request" panel with a GET method selected. The URL is set to `http://localhost:61235/api/v1/products/message?name1=Seb&name2=Max`. Below the URL, under the "Authorization" tab, it says "No Auth". The "Body" tab is selected, showing the response body as:

```
1 {
  "Message": "No HTTP resource was found that matches the request URI 'http://localhost:61235/api/v1/products/message?name1=Seb&name2=Max'.",
  "MessageDetail": "No action was found on the controller 'Products' that matches the request."
```

At the bottom of the Postman window, the status bar shows "Status: 404 Not Found Time: 26 ms Size: 672 B". The taskbar at the bottom of the screen shows various icons, including a file explorer, browser, and system tray.

Get All Products

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A search bar at the top right contains the URL 'http://localhost:61235/api/v1/products'. Below the search bar, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'JSON' tab is active, displaying a JSON array of three products. The products are as follows:

```
1 [
2   {
3     "Id": 1,
4     "Name": "Tomato Soup",
5     "Category": "Groceries",
6     "Price": 1.0
7   },
8   {
9     "Id": 2,
10    "Name": "Yo-yo",
11    "Category": "Toys",
12    "Price": 3.75
13  },
14  {
15    "Id": 3,
16    "Name": "Hammer",
17    "Category": "Hardware",
18    "Price": 16.99
19  }
20 ]
```

The bottom status bar shows the date and time as '14 May 2020 2:44 PM'.

Get Product

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A search bar at the top right contains the URL 'http://localhost:61235/api/v1/products/2'. Below the search bar, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'JSON' tab is active, displaying a single product object. The product is as follows:

```
1 {
2   "Id": 2,
3   "Name": "Yo-yo",
4   "Category": "Toys",
5   "Price": 3.75
6 }
```

The bottom status bar shows the date and time as '14 May 2020 2:45 PM'.

ID < 2

The screenshot shows the Postman application interface. In the center, there is a request card for a GET request to `http://localhost:61235/api/v1/products/1`. The status bar at the bottom indicates a 404 Not Found status with a time of 654 ms and a size of 5.22 KB. Below the request card, three sections provide troubleshooting information: 'Most likely causes', 'Things you can try', and 'Detailed Error Information'. The 'Most likely causes' section lists: 'The directory or file specified does not exist on the Web server.', 'The URL contains a typographical error.', and 'A custom filter or module, such as URLScan, restricts access to the file.' The 'Things you can try' section suggests: 'Create the content on the Web server.', 'Review the browser URL.', and 'Check the failed request tracing log and see which module is calling SetStatus. For more information, click [here](#)'. The 'Detailed Error Information' section is collapsed.

ID does not exist

This screenshot is nearly identical to the previous one, showing the same Postman interface and request card for `http://localhost:61235/api/v1/products/10`. The status bar at the bottom shows a 404 Not Found status with a time of 15 ms and a size of 387 B. The troubleshooting sections ('Most likely causes', 'Things you can try', and 'Detailed Error Information') are also present, providing the same diagnostic advice. The 'Detailed Error Information' section is collapsed.

Get Products by Category

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A search bar at the top right contains the URL 'http://localhost:61235/api/v2/products?category=Toys'. Below the search bar, there are several tabs: POST https..., POST https..., DEL https..., POST https..., POST https..., and GET http://localhost:61235/api/v2/products?category=Toys. The 'GET' tab is highlighted. To the right of the URL, there are 'Send' and 'Save' buttons. The main workspace shows a history of requests on the left and the response details on the right. The response status is 200 OK, time is 1785 ms, size is 480 B, and it includes a 'Save Response' button. The response body is displayed in JSON format:

```
1 {
2   "Id": 2,
3   "Name": "Yo-yo",
4   "Category": "Toys",
5   "Price": 3.75
6 }
```

Add Product

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A search bar at the top right contains the URL 'http://localhost:61235/api/v2/products'. Below the search bar, there are several tabs: POST https..., POST https..., DEL https..., POST https..., POST https..., and POST http://localhost:61235/api/v2/products. The 'POST' tab is highlighted. To the right of the URL, there are 'Send' and 'Save' buttons. The main workspace shows a history of requests on the left and the response details on the right. The response status is 201 Created, time is 757 ms, size is 536 B, and it includes a 'Save Response' button. The response body is displayed in JSON format:

```
1 {
2   "Id": 4,
3   "Name": "Batman",
4   "Category": "Toys",
5   "Price": 10.0
6 }
```

Edit Product

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. The main workspace shows an 'Untitled Request' with a 'PUT' method selected. The URL is set to `http://localhost:61235/api/v2/products/4`. The 'Body' tab is active, showing JSON data to update a product:

```
1 {  
2   "name": "Batman",  
3   "Category": "Toys",  
4   "price": "50"  
5 }
```

Below the request, the response details are shown: Status: 204 No Content, Time: 684 ms, Size: 365 B. The Windows taskbar at the bottom indicates the current date and time as 14 May 2020, 3:46 PM.

Delete Product

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. The main workspace shows an 'Untitled Request' with a 'DELETE' method selected. The URL is set to `http://localhost:61235/api/v2/products/4`. The 'Body' tab is active, showing a single digit '1'.

Below the request, the response details are shown: Status: 204 No Content, Time: 144 ms, Size: 365 B. The Windows taskbar at the bottom indicates the current date and time as 14 May 2020, 3:50 PM.

Get All Products (After Delete)

The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'My Workspace', 'Invite', 'Upgrade', and various icons. The left sidebar displays a history of API interactions, including a successful 'GET' request to 'http://localhost:61235/api/v2/products'. The main workspace shows a 'GET' request to the same URL, with the response body displayed in JSON format:

```
1  [
2   {
3     "Id": 1,
4     "Name": "Tomato soup",
5     "Category": "Groceries",
6     "Price": 1.39
7   },
8   {
9     "Id": 2,
10    "Name": "Yo-yo",
11    "Category": "Toys",
12    "Price": 3.75
13  },
14  {
15    "Id": 3,
16    "Name": "Hammer",
17    "Category": "Hardware",
18    "Price": 16.99
19  }
20 ]
```

The bottom taskbar shows various system icons and the date/time: '14 May 2020 3:50 PM'.

Practical 3 Part 1

API Documentation

1. Underposting

- a. Add a product
- b. Method: POST
- c. URL: api/v3/products
- d. Params:
 - i. string category
 - ii. decimal price

2. Overposting

- a. Add a product
- b. Method: POST
- c. URL: api/v3/products
- d. Params:
 - i. string name
 - ii. string category
 - iii. decimal price
 - iv. decimal discount

Postman Screenshots

Underposting

The screenshot shows the Postman interface with the following details:

- Header Bar:** Postman, File, Edit, View, Help, My Workspace, Invite, Upgrade.
- Toolbar:** New, Import, Runner, POST http..., GET http..., PUT http..., DELETE http..., PATCH http..., HEAD http..., No Environment, Send, Save, Beautify.
- Request Section:** Method: POST, URL: http://localhost:61963/api/v3/products, Body type: JSON. Request body:

```
1 {
2   "category": "Toys",
3   "price": -12
4 }
```

- Response Section:** Status: 400 Bad Request, Time: 6.09 s, Size: 586 B. Response body:

```
1 {
2   "Message": "The request is invalid.",
3   "ModelState": {
4     "item.Name": [
5       "The Name field is required."
6     ],
7     "item.Price": [
8       "The field Price must be between 0 and 100."
9     ]
10  }
11 }
```

Overposting

The screenshot shows the Postman application interface. In the center, there is a POST request to `http://localhost:61963/api/v3/products`. The request body contains the following JSON:

```
1 "name": "Ultraman",
2 "category": "Toys",
3 "price": 12,
4 "Discount": 12}
```

The response status is 201 Created, and the response body is:

```
1 {
2   "Id": 4,
3   "Name": "Ultraman",
4   "Category": "Toys",
5   "Price": 12.0
6 }
```

The left sidebar shows a history of requests made today and yesterday.

Task 3

Web API 2 Security Authentication and Authorization using Bearer Token

API Documentation

1. Get Values
 - a. Gets a greeting when user log in in String format
 - b. Method: GET
 - c. URL: api/values
2. Register
 - a. Registers an account
 - b. Method: POST
 - c. URL: api/Account/Register
 - d. Params
 - i. String email
 - ii. String password
 - iii. String confirmPassword
3. Login
 - a. Gets a Token to authorize the user using API
 - b. Method: POST
 - c. URL: api/Token
 - d. Params
 - i. String username
 - ii. String password
 - iii. String grant_type
4. Logout
 - a. Logs out the user
 - b. Method: POST
 - c. URL: api/Account/Logout

Postman Screenshots

Get Values

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'File', 'Edit', 'View', and 'Help' are visible. Below the bar, there are buttons for 'New', 'Import', 'Runner', and 'My Workspace'. A 'No Environment' dropdown is shown. On the right side, there are icons for refresh, search, and settings. The main workspace is titled 'Untitled Request'. It shows a GET request to 'https://localhost:44383/api/values'. The 'Headers' tab is selected, containing a single entry: 'Authorization' with value 'Bearer 7xh4o0FkEyUe170Qeb9uP_EsWNvmDLa1T...'. The 'Body' tab shows the response: 'Hello, alex@gmail.com.'. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 46 ms', and 'Size: 428 B'. The taskbar at the bottom of the screen shows various application icons.

Unauthorized:

The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', and 'Help'. Below the bar, there are buttons for 'New', 'Import', 'Runner', and 'My Workspace'. A 'No Environment' dropdown is present. The main workspace is titled 'Untitled Request'. It shows a GET request to 'https://localhost:44383/api/values'. The 'Headers' tab is selected, containing a single entry: 'Key' with value 'Value'. The 'Body' tab shows the response: 'Message": "Authorization has been denied for this request."'. The status bar at the bottom indicates 'Status: 401 Unauthorized', 'Time: 556 ms', and 'Size: 501 B'. The taskbar at the bottom of the screen shows various application icons.

Register

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A POST request is being made to `https://localhost:44383/api/Account/Register`. The 'Body' tab is active, showing three form fields: 'email' (alex@gmail.com), 'password' (P@ssw0rd), and 'confirmPassword' (P@ssw0rd). The 'Send' button is highlighted in blue. Below the request, the status bar indicates a 200 OK response with a time of 1159 ms and a size of 368 B. The bottom status bar shows the date and time as 26 May 2020, 11:15 AM.

Password doesn't match criteria:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. A POST request is being made to `https://localhost:44383/api/Account/Register`. The 'Body' tab is active, showing three form fields: 'email' (alex@gmail.com), 'password' (password), and 'confirmPassword' (password). The 'Send' button is highlighted in blue. Below the request, the status bar indicates a 400 Bad Request response with a time of 158 ms and a size of 653 B. The bottom status bar shows the date and time as 26 May 2020, 11:12 AM. The JSON response body is visible in the preview pane, showing validation errors for the password fields.

Password length < 6:

The screenshot shows the Postman application interface. A POST request is being made to `https://localhost:44383/api/Account/Register`. The request body is set to `x-www-form-urlencoded` and contains three parameters: `email` (value: `alex@gmail.com`), `password` (value: `passw`), and `confirmPassword` (value: `passw`). The response status is 400 Bad Request, indicating the password is too short. The JSON response message is: "Message": "The request is invalid.", "ModelState": { "model.Password": ["The Password must be at least 6 characters long."] }

Invalid Email:

The screenshot shows the Postman application interface. A POST request is being made to `https://localhost:44383/api/Account/Register`. The request body is set to `x-www-form-urlencoded` and contains three parameters: `email` (value: `alex`), `password` (value: `P@ssw0rd`), and `confirmPassword` (value: `P@ssw0rd`). The response status is 400 Bad Request, indicating the email is invalid. The JSON response message is: "Message": "The request is invalid.", "ModelState": { "": ["Email 'alex' is invalid."] }

Login

Success:

The screenshot shows the Postman application interface. In the center, a POST request is being made to `https://localhost:44383/Token`. The request body contains three fields: `username` (alex@gmail.com), `password` (P@ssw0rd), and `grant_type` (password). The response status is 200 OK, and the response body is a JSON token object.

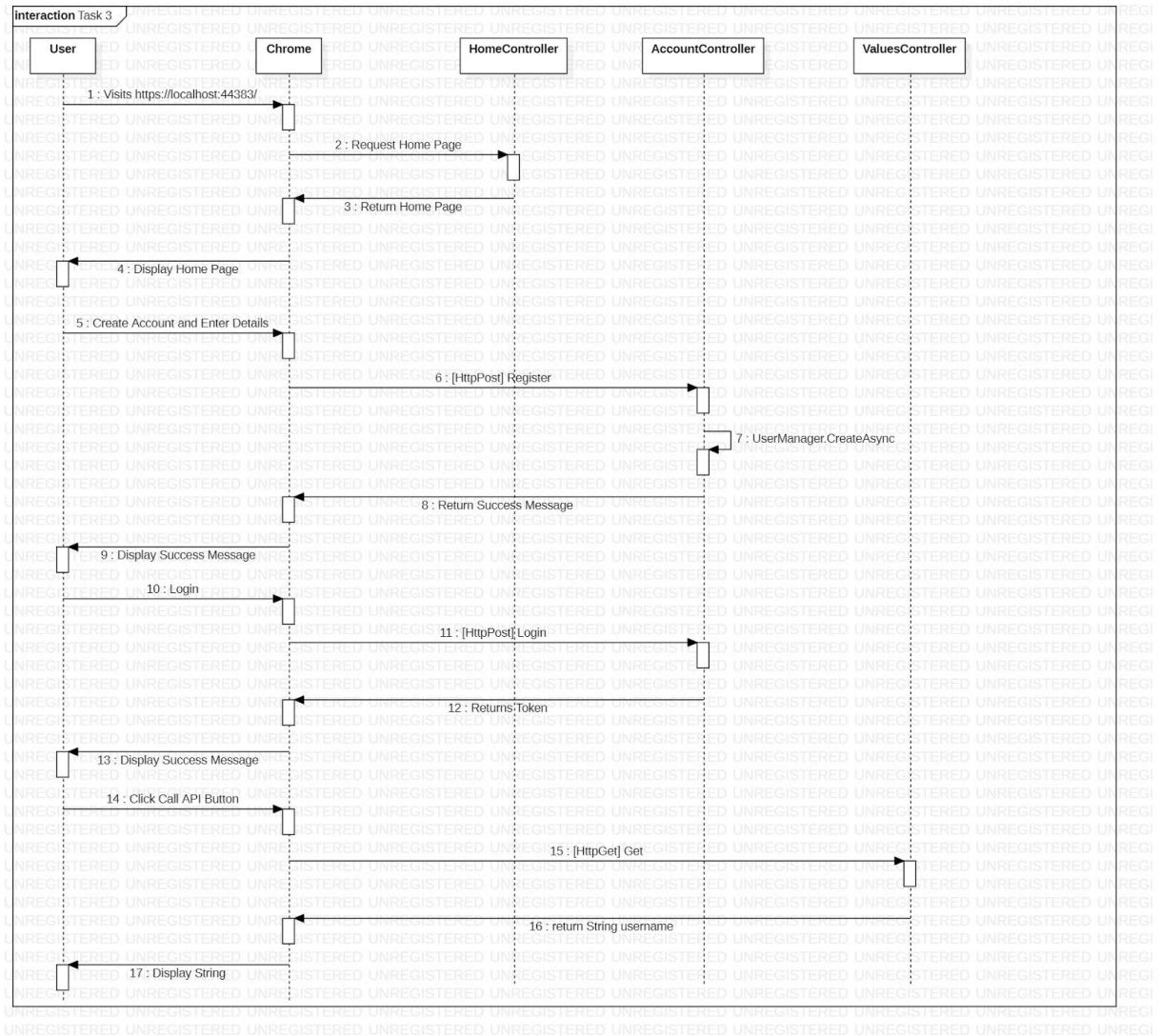
```
1 "access_token":  
2     "Lwdov39MjBRArpvf40bHClws-fg6MbWeHaPoQc3z_0IKqLEDEJezg0D08fGxMqwqtZ4Xy17ojnQYyWdASyqmewYK08-hivNliiT8_iuajcDbJRxwKtwGG2tWw0UD  
3     Qet5C08cgOHwM!af712PwclFfmwykrv-n9g1q5VB7gFlPARq1Bupj0qrVBSU_Hx70MsIr4bGfL5Lfset5rZ40cJ2kh-dy7_FyyhdzuL370Qah-BTuq1gfmbXt  
4     zK0Ktp6ftfU66ByKwG_5n-6rK2HjgTqfeho3kRdv4VViwxHFiYXUcbatrwgQj7bu02ANyrlBL7cGhzXvxaVatokdrL2-qv93R48Kgy0kCAqf2m1yTwdwHg  
5     L5Q0Hap7pas435Xv_yxxsqvZwSfVIALrEhwrRXeiebHJM3u4XB8izBSAust18e2LYbcjKpcPjyMxaarPHSNRZh0n9Qo4jxyZQyM0dsSuq0G520ALG10J0LCcENgm  
6     TLvmlijkoB7x3,  
7     "token_type": "bearer",  
8     "expires_in": 1209599,
```

Logout

Success:

The screenshot shows the Postman application interface. A POST request is being made to `https://localhost:44383/api/Account/Logout`. The request includes an Authorization header with the value `Bearer Lwdov39MjBRArpvf40bHClws-fg6MbWe...`. The response status is 200 OK, and the response body is empty.

Diagram



Task 4

Talents Search Functions & Secured Talent Search Web API

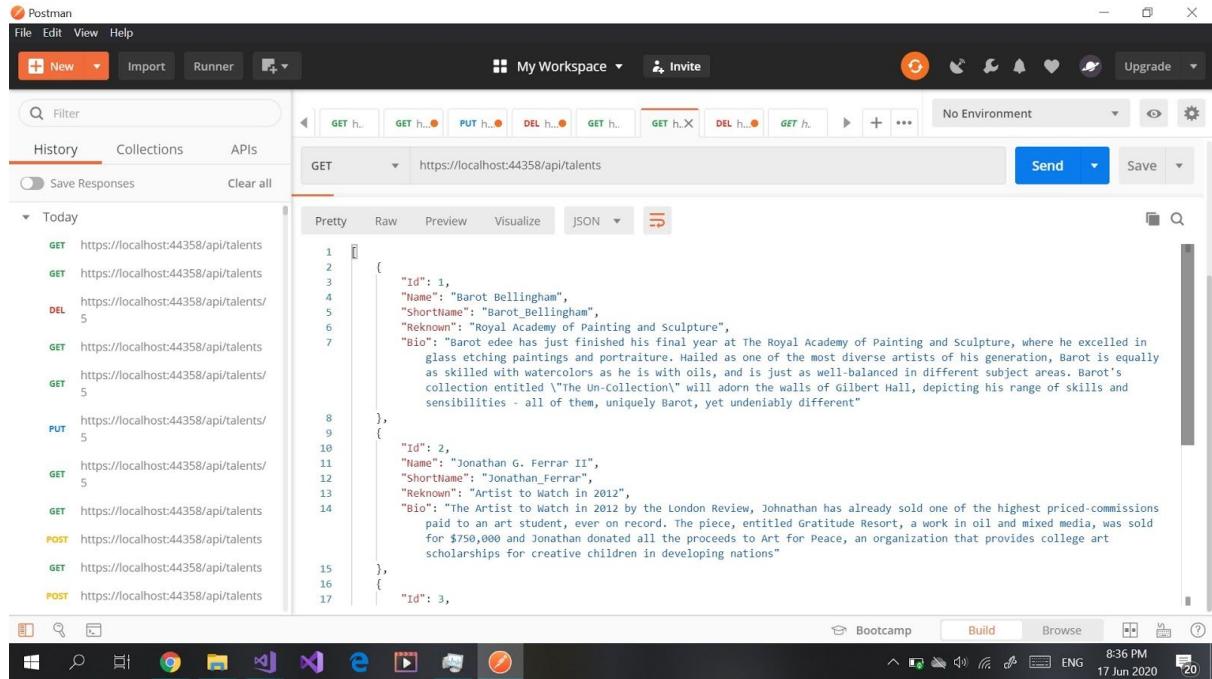
API Documentation

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
 - 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:

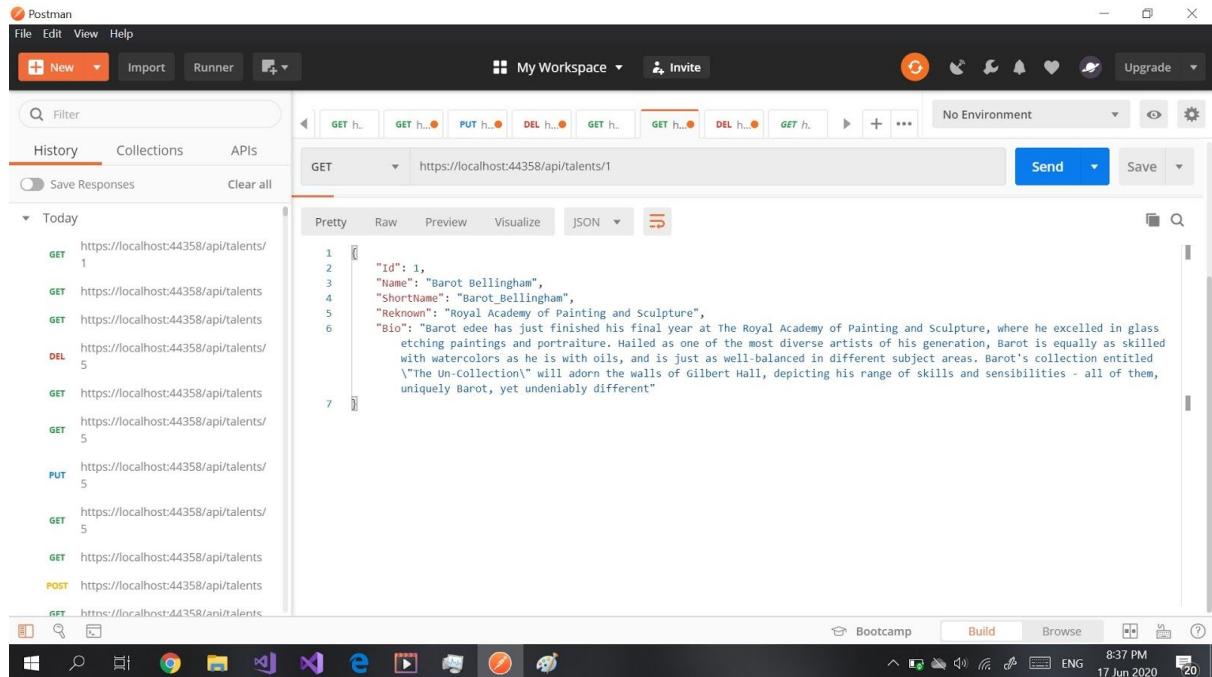


The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'New', 'Import', 'Runner', and 'My Workspace'. The main workspace shows a 'GET' request to 'https://localhost:44358/api/talents'. The response is displayed in 'Pretty' JSON format, listing two talents: Barot Bellingham and Jonathan G. Ferrar II. Both entries include their ID, name, short name, recognition, and bio.

```
1  [
2   {
3     "Id": 1,
4     "Name": "Barot_Bellingham",
5     "Shortname": "Barot_Bellingham",
6     "Rekrown": "Royal Academy of Painting and Sculpture",
7     "Bio": "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"
8   },
9   {
10    "Id": 2,
11    "Name": "Jonathan G. Ferrar II",
12    "Shortname": "Jonathan_Ferrar",
13    "Rekrown": "Artist to Watch in 2012",
14    "Bio": "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 Johnathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations"
15  },
16  {
17    "Id": 3,
```

Get Talent

Success:



The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'New', 'Import', 'Runner', and 'My Workspace'. The main workspace shows a 'GET' request to 'https://localhost:44358/api/talents/1'. The response is displayed in 'Pretty' JSON format, showing the details for talent ID 1, Barot Bellingham.

```
1  [
2   {
3     "Id": 1,
4     "Name": "Barot_Bellingham",
5     "Shortname": "Barot_Bellingham",
6     "Rekrown": "Royal Academy of Painting and Sculpture",
7     "Bio": "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"
7  ]
```

Add Talent

Success:

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. The main workspace shows a POST request to 'https://localhost:44358/api/talents'. The request body contains the following JSON data:

```
1 "Id": 5,
2 "Name": "Michael Kim",
3 "ShortName": "Michael_Kim",
4 "Rekrown": "Best F1 Driver",
5 "Bio": "Michael Kim is the best F1 driver in the world!"
```

The response status is 200 OK, time 239 ms, and size 563 B. The bottom status bar shows the date and time as 17 Jun 2020 8:43 PM.

Result

The screenshot shows the Postman application interface. In the top navigation bar, 'My Workspace' is selected. The main workspace shows a GET request to 'https://localhost:44358/api/talents'. The response body displays a list of talents in JSON format, with lines numbered 20 through 37:

```
20 "Rekrown": "New York University",
21 "Bio": "Hillary is a sophomore art sculpture student at New York University, and has already won all the major international
22 prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris
23 Award. Hillary's GAC exhibit features 25 abstract watercolor paintings that contain only water images including waves,
24 deep sea, and river."
25 },
26 {
27 "Id": 4,
28 "Name": "Hassum Harrod",
29 "Shortname": "Hassum_Harrod",
30 "Rekrown": "Art College in New Dehli",
31 "Bio": "The Art College in New Dehli has sponsored Hassum on scholarship for his entire undergraduate career at the
32 university, seeing great promise in his contemporary paintings of landscapes - that use equal parts muted and vibrant
33 tones, and are almost a contradiction in art. Hassum will be speaking on \"The use and absence of color in modern art\""
34 during Thursday's agenda."
35 },
36 {
37 "Id": 5,
38 "Name": "Michael Kim",
39 "Shortname": "Michael_Kim",
40 "Rekrown": "Best F1 Driver",
41 "Bio": "Michael Kim is the best F1 driver in the world!"
```

The bottom status bar shows the date and time as 17 Jun 2020 8:45 PM.

Edit Talent

The screenshot shows the Postman application interface. In the top navigation bar, 'File', 'Edit', 'View', and 'Help' are visible. Below the bar, there are buttons for 'New', 'Import', 'Runner', and a search icon. The 'My Workspace' button is highlighted. On the right side, there are icons for 'Invite', 'Upgrade', and a gear settings icon.

The main workspace shows a history of API requests on the left. A specific PUT request is selected for editing:

```
PUT https://localhost:44358/api/talents/5
```

The request details panel shows the method as 'PUT' and the URL as 'https://localhost:44358/api/talents/5'. The body tab contains a table with key-value pairs:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Name	Michael Kim	
<input checked="" type="checkbox"/> ShortName	Michael_Kim	
<input checked="" type="checkbox"/> Rekrown	Worst F1 Driver	
<input checked="" type="checkbox"/> Bio	Michael Kim is the worst and slowest F1 driver in t ...	
Key	Value	Description

The response panel shows the status as '200 OK', time as '227 ms', and size as '577 B'. The response body is displayed in JSON format:

```

1   {
2     "id": 5,
3     "Name": "Michael Kim",
4     "ShortName": "Michael_Kim",
5     "Rekrown": "Worst F1 Driver",
6     "Bio": "Michael Kim is the worst and slowest F1 driver in the world!"
7   }

```

The bottom of the window shows the Windows taskbar with various pinned icons like File Explorer, Task View, and Edge. The system tray indicates the date as '17 Jun 2020' and the time as '8:45 PM'.

Result

The screenshot shows the Postman application interface again. The top navigation bar and workspace are identical to the previous screenshot.

A new GET request is selected for viewing:

```
GET https://localhost:44358/api/talents
```

The response panel shows the status as '200 OK', time as '227 ms', and size as '577 B'. The response body is displayed in JSON format, showing multiple talent records:

```

20  [
21    {
22      "Rekrown": "New York University",
23      "Bio": "Hillary is a sophomore art sculpture student at New York University, and has already won all the major international
24      prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris
25      Award. Hillary's CAC exhibit features 25 abstract watercolor paintings that contain only water images including waves,
26      deep sea, and river."
27    },
28    {
29      "Id": 4,
30      "Name": "Hassum Harrod",
31      "Shortname": "Hassum_Harrod",
32      "Rekrown": "Art College in New Dehli",
33      "Bio": "The Art College in New Dehli has sponsored Hassum on scholarship for his entire undergraduate career at the
34      university, seeing great promise in his contemporary paintings of landscapes - that use equal parts muted and vibrant
35      tones, and are almost a contradiction in art. Hassum will be speaking on \'The use and absence of color in modern art\'"
36    },
37    {
38      "Id": 5,
39      "Name": "Michael Kim",
40      "Shortname": "Michael_Kim",
41      "Rekrown": "Worst F1 Driver",
42      "Bio": "Michael Kim is the worst and slowest F1 driver in the world!"
43    }
44  ]

```

The bottom of the window shows the Windows taskbar and system tray, indicating the date as '17 Jun 2020' and the time as '8:47 PM'.

Delete Talent

Success:

The screenshot shows the Postman application interface. In the center, there is a "Untitled Request" panel with a "DELETE" method selected and the URL `https://localhost:44358/api/talents/5`. Below the URL, under the "Body" tab, the response is displayed as a JSON object:

```
1  "Deleted 5"
```

At the bottom right of the Postman window, the status bar shows "Status: 200 OK Time: 23 ms Size: 435 B".

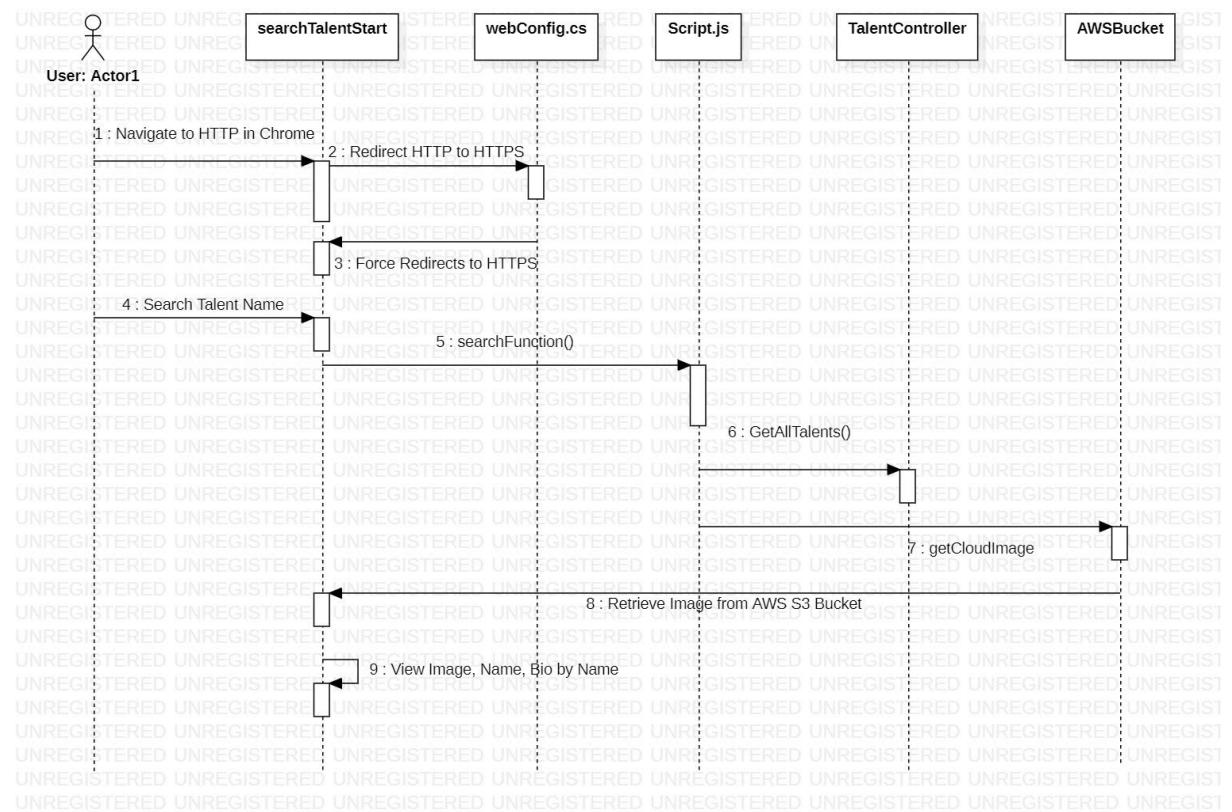
Result

The screenshot shows the Postman application interface. In the center, there is a "GET" method selected and the URL `https://localhost:44358/api/talents`. The response is displayed as a JSON array of talents:

```
15  },
16  {
17    "Id": 3,
18    "Name": "Hillary Hewitt Goldwynn-Post",
19    "Shortname": "Hillary_Goldwynn",
20    "Rekrown": "New York University",
21    "Bio": "Hillary is a sophomore art sculpture student at New York University, and has already won all the major international prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris Award. Hillary's CAC exhibit features 25 abstract watercolor paintings that contain only water images including waves, deep sea, and river."
22  },
23  {
24    "Id": 4,
25    "Name": "Hassum Harrod",
26    "Shortname": "Hassum_Harrod",
27    "Rekrown": "Art College in New Dehli",
28    "Bio": "The Art College in New Dehli has sponsored Hassum on scholarship for his entire undergraduate career at the university, seeing great promise in his contemporary paintings of landscapes - that use equal parts muted and vibrant tones, and are almost a contradiction in art. Hassum will be speaking on \"The use and absence of color in modern art\" during Thursday's agenda."
29  }
30 ]
```

At the bottom right of the Postman window, the status bar shows "Status: 200 OK Time: 23 ms Size: 435 B".

Diagram



Task 5

Store Talents Photo in AWS S3 & Generate bit.ly API

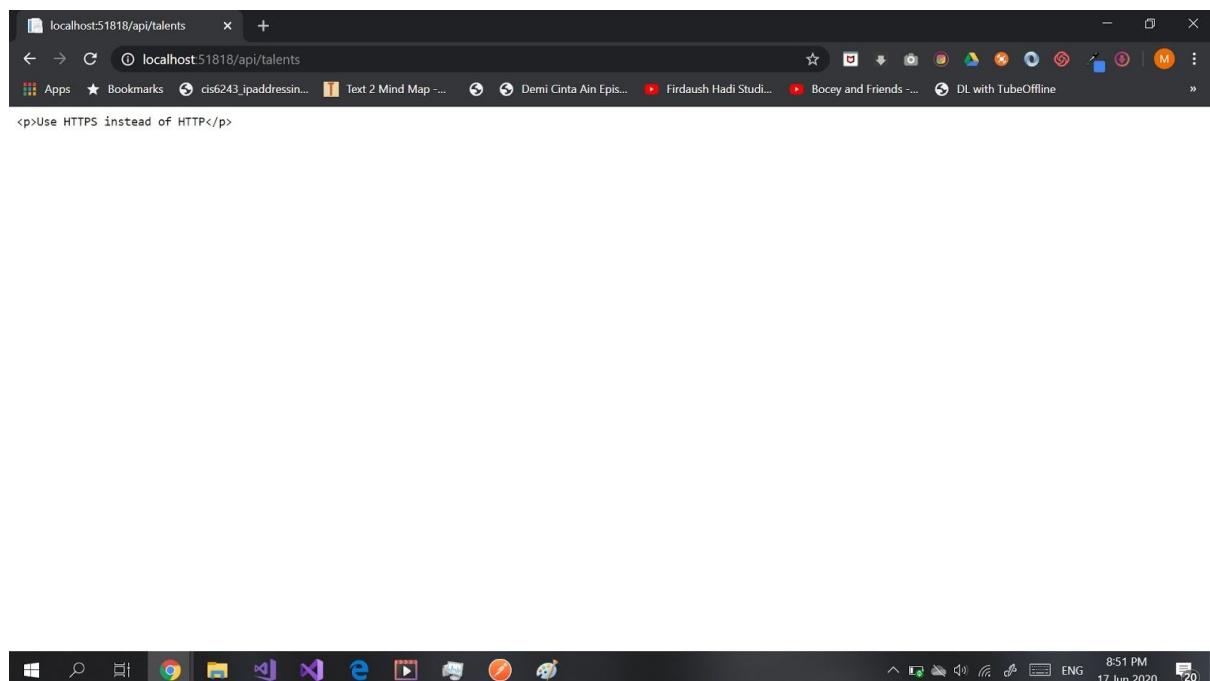
API Documentation

[Same as Task 4]

Postman Screenshots

Get list of Talents using HTTP

The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'New', 'Import', 'Runner', 'My Workspace', 'Invite', 'Upgrade', and various icons. The main workspace has tabs for 'History' (selected), 'Collections', and 'APIs'. A 'Save Responses' button is also present. In the center, a 'GET' request is selected, with the URL 'http://localhost:51818/api/talents' entered. Below the URL, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text'. The response pane shows a single line of text: '<p>Use HTTPS instead of HTTP</p>'. On the left, a sidebar lists several recent API requests, mostly GET requests to 'http://localhost:51818/api/talents' and 'https://localhost:44358/api/talents'. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Google Chrome, and Microsoft Edge.

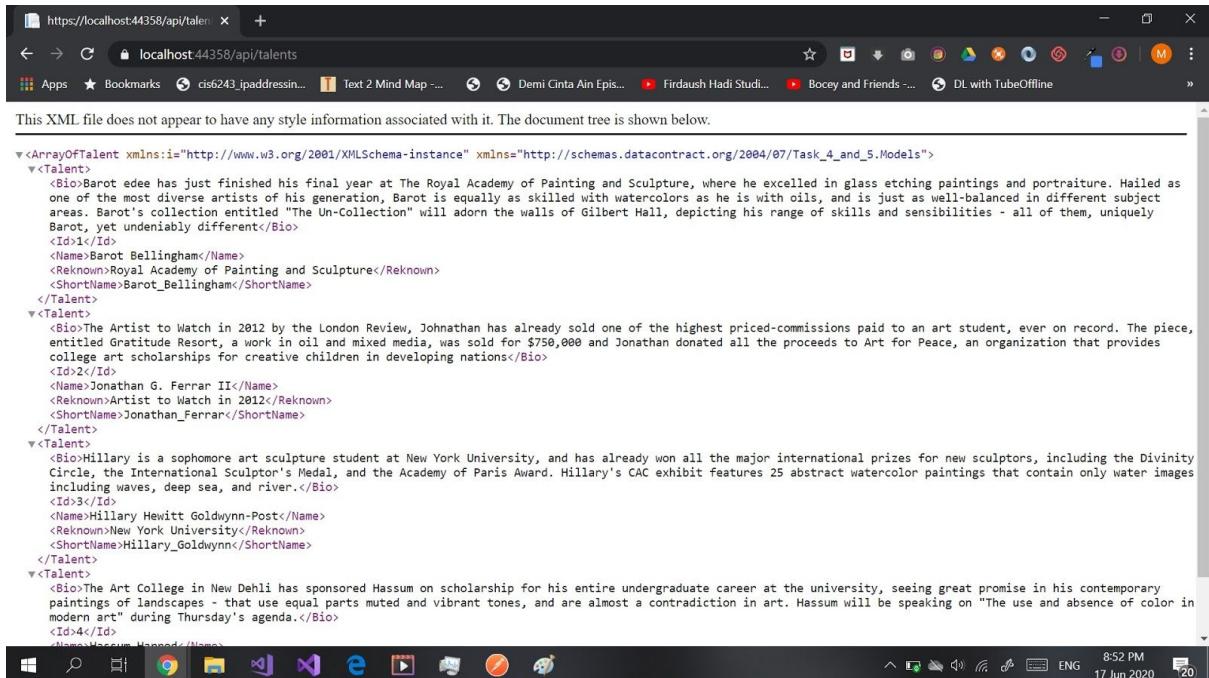


Get list of Talents using HTTPS

A screenshot of the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'New', 'Import', 'Runner', 'My Workspace', 'Invite', 'Upgrade', and various status indicators like 'No Environment' and 'Send'. The main workspace shows a 'History' tab with several API requests listed. A specific 'GET' request is selected, with the URL 'https://localhost:44358/api/talents' entered in the 'Method' and 'URL' fields. The response body is displayed in 'Pretty' JSON format:

```
1 {
  "Id": 1,
  "Name": "Barot_Bellingham",
  "Shortname": "Barot_Bellingham",
  "Rekrown": "Royal Academy of Painting and Sculpture",
  "Bio": "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"
},
{
  "Id": 2,
  "Name": "Jonathan_G_Ferrari_II",
  "Shortname": "Jonathan_Ferrari",
  "Rekrown": "Artist to Watch in 2012",
  "Bio": "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 Jonathon donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations"
},
{
  "Id": 3,
```

The bottom of the screen shows the Windows taskbar with various pinned icons and system status information: 'Bootcamp', 'Build', 'Browse', '8:52 PM', 'ENG', '17 Jun 2020', and a battery icon.



```
<ArrayOfTalent xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/Task_4_and_5.Models">
    <Talent>
        <Bio>Barot eden 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.</Bio>
        <Id>1</Id>
        <Name>Barot Bellingham</Name>
        <Reknown>Royal Academy of Painting and Sculpture</Reknown>
        <ShortName>Barot_Bellingham</ShortName>
    </Talent>
    <Talent>
        <Bio>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.</Bio>
        <Id>2</Id>
        <Name>Johnathan G. Ferrar II</Name>
        <Reknown>Artist to Watch in 2012</Reknown>
        <ShortName>Johnathan_Ferrar</ShortName>
    </Talent>
    <Talent>
        <Bio>Hillary is a sophomore art sculpture student at New York University, and has already won all the major international prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris Award. Hillary's CAC exhibit features 25 abstract watercolor paintings that contain only water images including waves, deep sea, and river.</Bio>
        <Id>3</Id>
        <Name>Hillary Hewitt Goldwynn-Post</Name>
        <Reknown>New York University</Reknown>
        <ShortName>Hillary_Goldwynn</ShortName>
    </Talent>
    <Talent>
        <Bio>The Art College in New Dehli has sponsored Hassum on scholarship for his entire undergraduate career at the university, seeing great promise in his contemporary paintings of landscapes - that use equal parts muted and vibrant tones, and are almost a contradiction in art. Hassum will be speaking on "The use and absence of color in modern art" during Thursday's agenda.</Bio>
        <Id>4</Id>
        <Name>Hascum_Hassum</Name>
    </Talent>

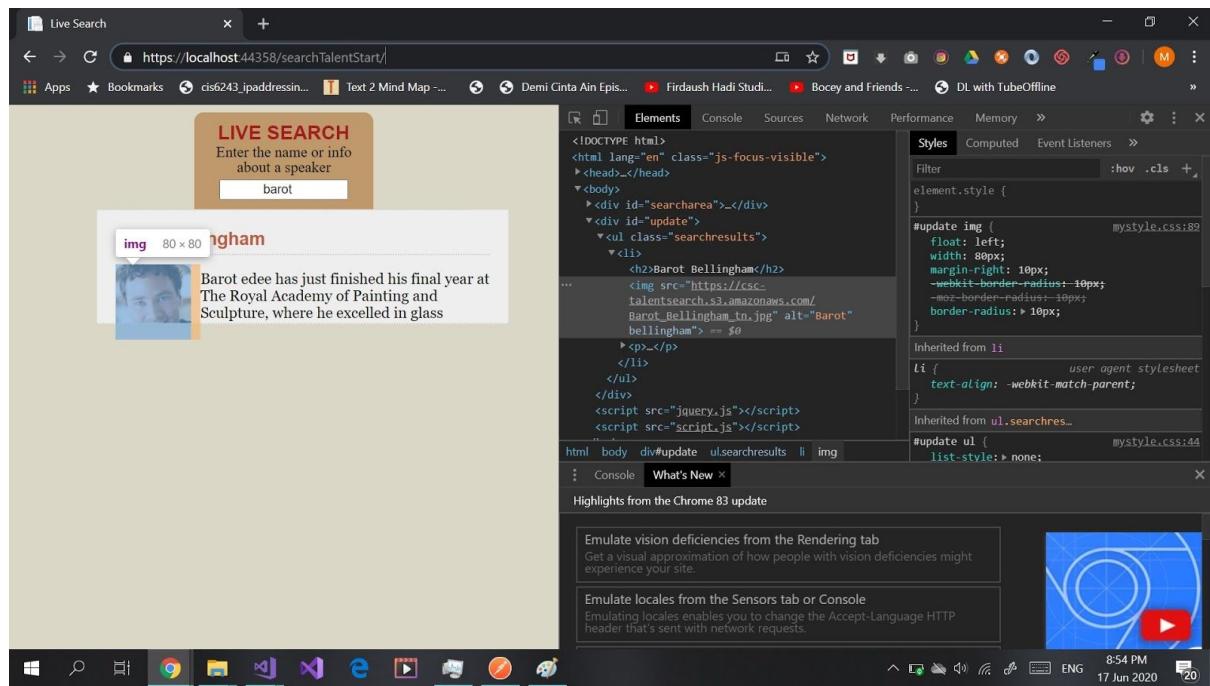
```

Store Talents in AWS S3 Bucket

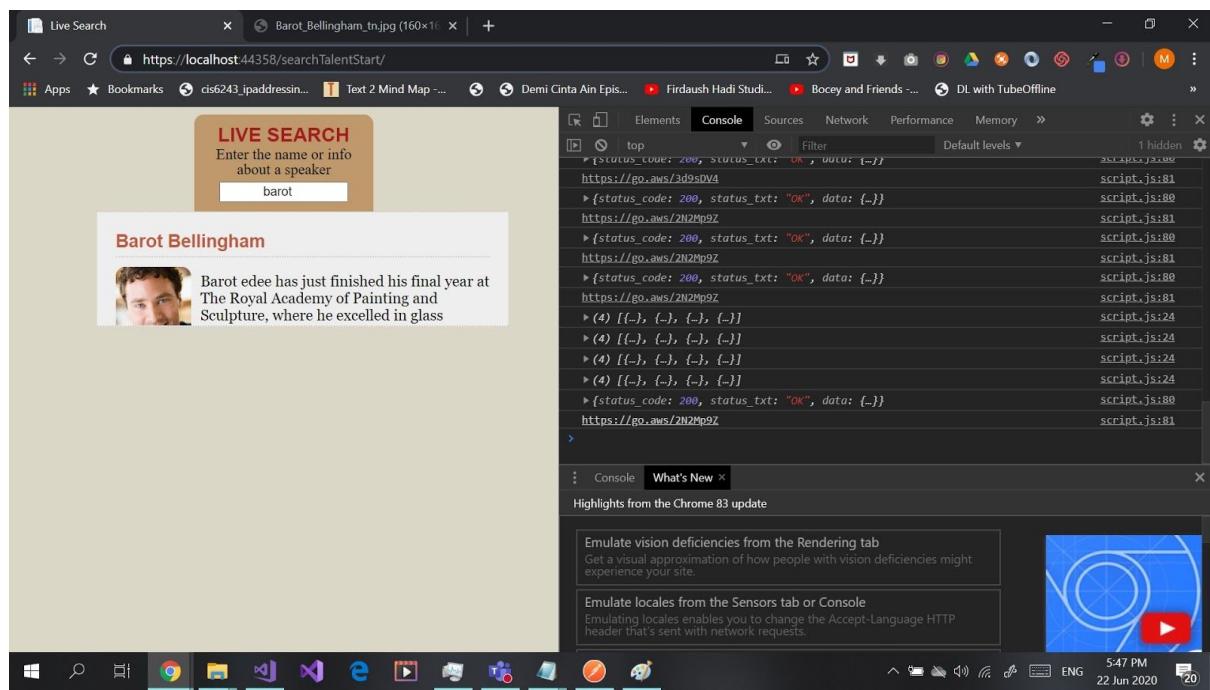
The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, the URL 'vocstartsoft/user353764=muh...', 'Global' dropdown, and 'Support' dropdown. Below the navigation bar, the path 'Amazon S3 > csc-talentsearch' is shown. The main area displays the 'csc-talentsearch' bucket. A navigation bar at the top of the bucket view includes 'Overview' (selected), 'Properties', 'Permissions', 'Management', and 'Access points'. Below this is a search bar with the placeholder 'Type a prefix and press Enter to search. Press ESC to clear.' Underneath the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. To the right of these buttons, it says 'US East (N. Virginia)' with a refresh icon. A table follows, showing 11 items. The columns are 'Name' (with a checkbox header), 'Last modified', 'Size', and 'Storage class'. The items listed are:

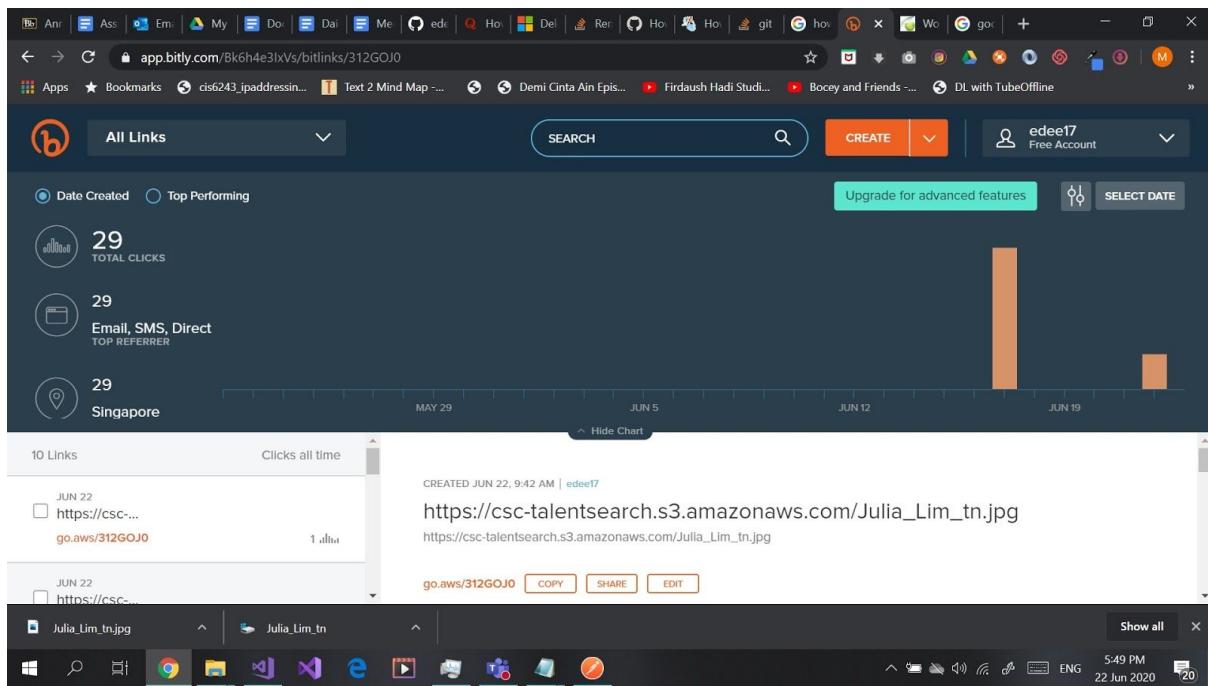
Name	Last modified	Size	Storage class
Barot_Bellingham_tn.jpg	Jun 9, 2020 12:17:15 PM GMT+0800	12.5 KB	Standard
Ben_Daly_tn.jpg	Jun 16, 2020 9:41:34 PM GMT+0800	821.7 KB	Standard
Constance_Smith_tn.jpg	Jun 9, 2020 12:17:15 PM GMT+0800	12.5 KB	Standard
Hassum_Harrod_tn.jpg	Jun 9, 2020 12:17:15 PM GMT+0800	11.1 KB	Standard
Hillary_Goldwynn_tn.jpg	Jun 9, 2020 12:17:15 PM GMT+0800	10.8 KB	Standard
Jennifer_Jerome_tn.jpg	Jun 9, 2020 12:17:16 PM GMT+0800	15.0 KB	Standard
Jonathan_Ferrari_tn.jpg	Jun 9, 2020 12:17:16 PM GMT+0800	13.0 KB	Standard
LaVonne_LaRue_tn.jpg	Jun 9, 2020 12:17:16 PM GMT+0800	12.2 KB	Standard
Michael_Kim_tn.jpg	Jun 17, 2020 8:41:11 PM GMT+0800	854.3 KB	Standard
Riley_Rewington_tn.jpg	Jun 9, 2020 12:17:16 PM GMT+0800	11.8 KB	Standard
Xhou_Ta_tn.jpg	Jun 9, 2020 12:17:15 PM GMT+0800	12.8 KB	Standard

At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Feedback' again. The footer contains the copyright notice '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

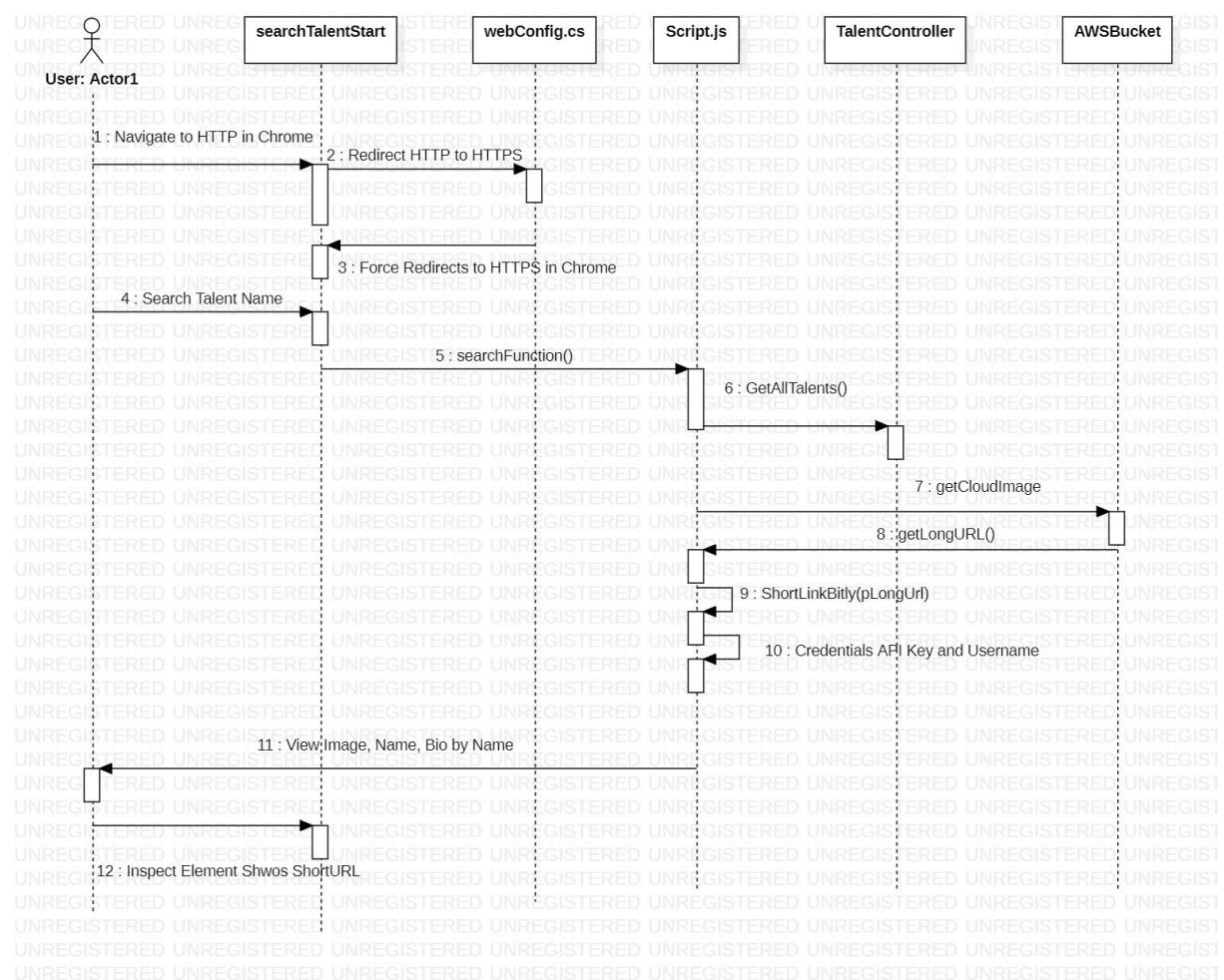


Generate bit.ly API





Diagram



Task 6

Develop a web client that accept subscription payment using Stripe billing subscription API & Use Stripe webhooks to update your database when the subscription plan changes

API Documentation

1. CustomersController
 - a. Get
 - i. Get all customers.
 - ii. Method: Get
 - iii. Url: /API/Customers/
 - b. GetByEmail
 - i. Get by email of customers
 - ii. Method: Post
 - iii. Url: /API/Customers/GetByEmail
 - iv. Data: FormData
2. SubscriptionsController
 - a. Get
 - i. Get all subscriptions
 - ii. Method: Get
 - iii. Url: /API/Subscriptions/
 - b. Get id
 - i. Get by id of subscription
 - ii. Method: Get
 - iii. Url: /API/Subscriptions?{id}
 - iv. Params:
 1. int id
3. 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

Screenshots

The screenshot shows the 'Customer Subscription' page of the StripeSubscribe application. At the top, there is a navigation bar with links for 'StripeSubscribe', 'Home', and 'Privacy'. Below the navigation, the main title 'Customer Subscription' is displayed. A prominent 'Add Customer' button is centered above a form field labeled 'Email'. To the right of the email field is a 'Submit' button. Below the form fields, there are three input fields labeled 'No.', 'Name', and 'Plan Id'. At the bottom of the page, a copyright notice reads '© 2020 - StripeSubscribe - Privacy'.

This screenshot shows the same 'Customer Subscription' page, but the 'Add Customer' form has been populated. The 'Email' field now contains 'boyo@gmail.com'. The 'Plan' section includes a credit card number '4242 4242 4242 4242', an expiration date '04 / 24', and a CVC '242'. Below the card details, there are two radio buttons for payment frequency: 'Monthly' (selected) and 'Yearly'. A 'Make Subscription' button is located at the bottom of the form. The rest of the page, including the navigation bar and footer, remains the same as the first screenshot.

Your subscription was successfully setup

© 2020 - StripeSubscribe - Privacy

Customer Subscription

[Add Customer](#)

Email

boyo@gmail.com

[Submit](#)

Customer ID:

cus_HXDSghUWHZc2RC

No. Name Plan Id

1 sub_HXDS2HCTzfe9vU price_1Gxk9eA5HN2CrOAmPEPjrx [Update](#)

© 2020 - StripeSubscribe - Privacy

Customer Subscription

[Add Customer](#)

Email

boyo@gmail.com

[Submit](#)

Customer ID:

cus_HXDSghUWHZc2RC

No. Name Plan Id

1 sub_HXDS2HCTzfe9vU price_1Gxk9eA5HN2CrOAmPEPjrx [Update](#)

Update Subscription

Plan

Monthly

Yearly

[Update Subscription](#)

© 2020 - StripeSubscribe - Privacy

Your Update is being processed.

[Click here to return to the Home page.](#)

Add a name

Home
Activate your account

- Payments
- Balances
- Customers
- Connected accounts
- Products
- Reports

Developers
Viewing test data
Settings

Search... / Feedback about this page?

Welcome, Muhammad—follow these steps to get started

Find the right integration for your business
Browse our docs for use cases, sample code, and developer tools or see options without code

[Browse docs →](#)

Get your test API keys

The email muhammadhusnimh.18@ichat.sp.edu.sg is verified

Activate your Stripe account

Get your live API keys

TEST DATA

Today

Gross volume	Yesterday	SGD Balance
\$35.00	\$30.00	\$62.62
12:55 PM		Estimated future payouts
12:00 AM	Now, 12:55 PM	11:59 PM

Payouts

Reports overview

Last 7 days Jun 20-Jun 26 compared to Previous period

Daily Hourly

Gross volume \$70.00	Net volume from sales \$62.62	New customers 14
Successful payments 10	Spend per customer \$7.77	Dispute activity 0%
Dispute count 0	High risk payments 0	New subscribers 0
MRR \$0.00	New trials 0	Trial conversion rate 0%
Lifetime value \$0.00	Revenue per subscriber \$0.00	Subscriber churn rate 0%
Churned revenue \$0.00	Subscriber retention by cohort	
	1 2 3 4 5 6 7 8 9 10 11 12	

The screenshot shows a software application interface with a sidebar on the left and a main content area on the right.

Left Sidebar:

- Add a name
- Home
- Activate your account
- Payments
- Balances
- Customers** (highlighted)
- Subscriptions
- Invoices
- Connected accounts
- Products
- Reports
- Developers
- Viewing test data
- Settings

Main Content Area:

Search bar: Search... /

TEST DATA

EMAIL	DESCRIPTION	DEFAULT SOURCE	CREATED
boyo@gmail.com	—	✓ ... 4242	Jun 26, 12:19 PM
yeetmail@email.com	—	✓ ... 4242	Jun 26, 2:06 AM
superman@hotmail.com	—	✓ ... 4242	Jun 25, 10:15 AM
a@a.com	—	✓ ... 4242	Jun 23, 5:36 PM

4 results

+ New

Previous

Add a name ▼

Home
Activate your account
Payments
Balances
Customers
Subscriptions
Invoices
Connected accounts
Products
Reports
Developers
Viewing test data
Settings

Search... Feedback about this page? ...

Customer
boyo@gmail.com

Created Jun 26, 12:19 PM Payments 1 payment for \$5.00

Timeline + Add note

- Changed subscription from Daily 5 to Daily 10 Jun 26, 2020, 12:20 PM
- Subscribed to Daily 5 Jun 26, 2020, 12:19 PM
- Customer was created Jun 26, 2020, 12:19 PM

Details Edit details

ACCOUNT INFORMATION		BILLING INFORMATION	
Name	No name	Send emails to	boyo@gmail.com
Email	boyo@gmail.com	Billing details	No details
ID	cus_H00SghUdH2c2RC	Next invoice number	0E8243BF-0092
Created	Jun 26, 2020, 12:19 PM	Tax IDs	No tax IDs
Language	English	Tax status	Taxable

Metadata Edit metadata

No metadata

Customer balance + Add balance adjustment

\$0.00 SGD

Payment methods + Add card ...

> Visa **** 4242 Default Expires April 2024

Payments + Add payment

AMOUNT	DESCRIPTION	CUSTOMER	DATE
\$5.00 SGD Succeeded ✓	Subscription creation	boyo@gmail.com	Jun 26, 2020, 12:19 PM ...

Subscriptions + Create subscription

Rubber Duck delivery Active Billing daily • Next invoice on Jun 27 for \$15.00

Invoices + Create invoice

AMOUNT	INVOICE NUMBER	CUSTOMER	DUEDUE	CREATED
\$5.00 SGD Paid 0E8243BF-0091	boyo@gmail.com	— Jun 26, 2020, 12:19 PM ...		

Pending invoice items + Create invoice item + Invoice now

AMOUNT	ITEM ID	DESCRIPTION	DATE
(\$4.99 SGD) cus_H00SghUdH2c2RC	sub_H00S2HCTzfe9vU	ii_1Gy93eA5HN82Cr0Ah12661Jt	Jun 26, 2020, 12:20 PM ...
\$9.99 SGD cus_H00SghUdH2c2RC	sub_H00S2HCTzfe9vU	ii_1Gy93eA5HN82Cr0Ah1e4K0	Jun 26, 2020, 12:20 PM ...

Upcoming invoice

View full invoice for \$15.00 on Jun 27 →

Coupon + Add coupon

No active coupon

Logs

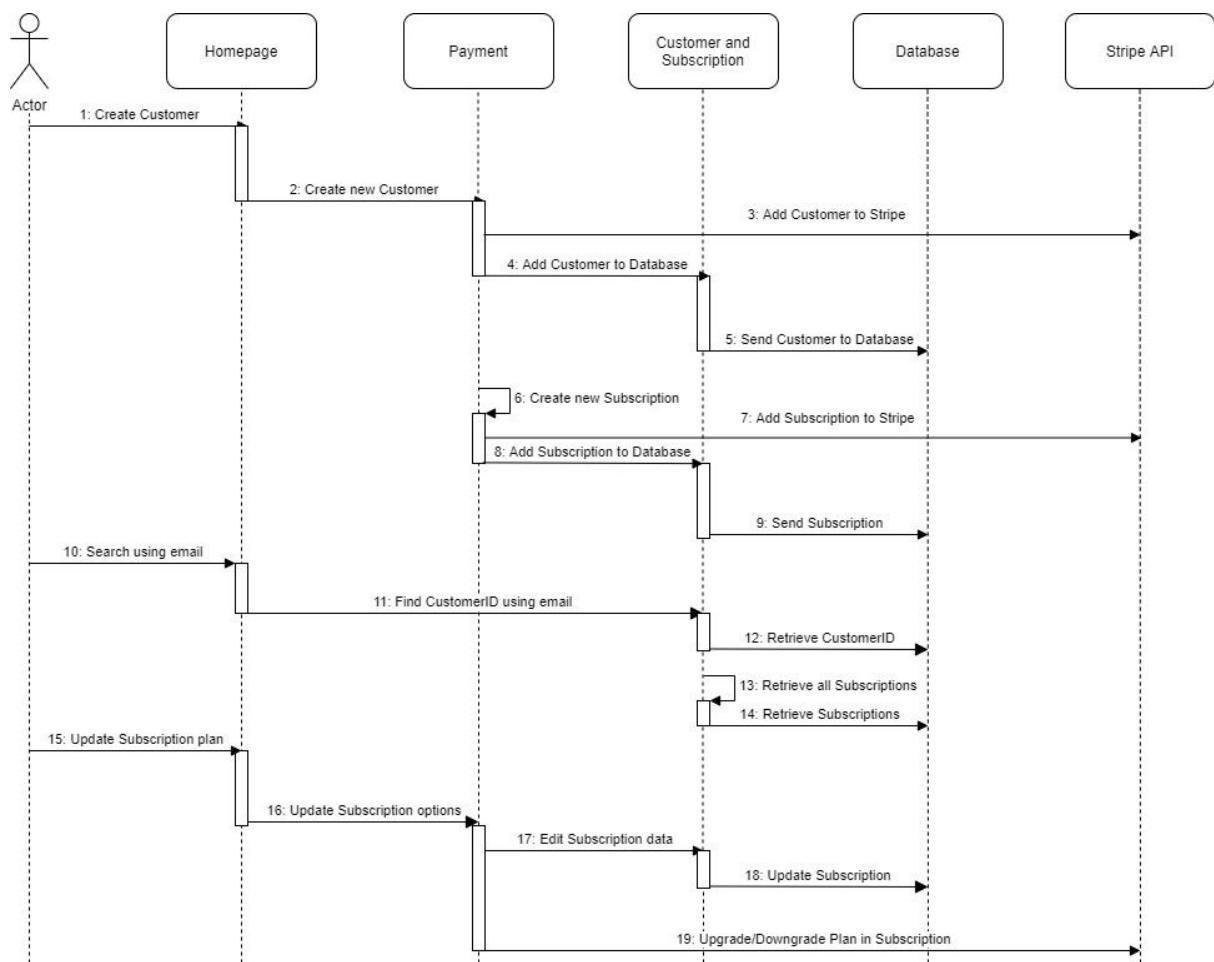
200 OK	POST /v1/subscriptions/sub_H00S2HCTzfe9vU	6/26/20, 12:20:57 PM
200 OK	POST /v1/subscriptions	6/26/20, 12:19:07 PM
200 OK	POST /v1/customers	6/26/20, 12:19:05 PM

Events

boyo@gmail.com upgraded to Daily 10 from Daily 5	6/26/20, 12:20:58 PM
A proration adjustment for \$9.99 SGD was created for boyo@gmail.com	6/26/20, 12:20:58 PM
A proration adjustment for -\$4.99 SGD was created for boyo@gmail.com	6/26/20, 12:20:58 PM
boyo@gmail.com subscribed to Daily 5	6/26/20, 12:19:09 PM
boyo@gmail.com's payment for an invoice for \$5.00 SGD succeeded	6/26/20, 12:19:09 PM
boyo@gmail.com's invoice for \$5.00 SGD was paid	6/26/20, 12:19:09 PM
A draft invoice for \$5.00 SGD to boyo@gmail.com was finalized	6/26/20, 12:19:09 PM
A draft invoice was created	6/26/20, 12:19:09 PM
boyo@gmail.com's details were updated	6/26/20, 12:19:09 PM
The payment pi_1Gy93eA5HN82Cr0Ah1e4K0 for \$5.00 SGD has succeeded	6/26/20, 12:19:09 PM

View more events

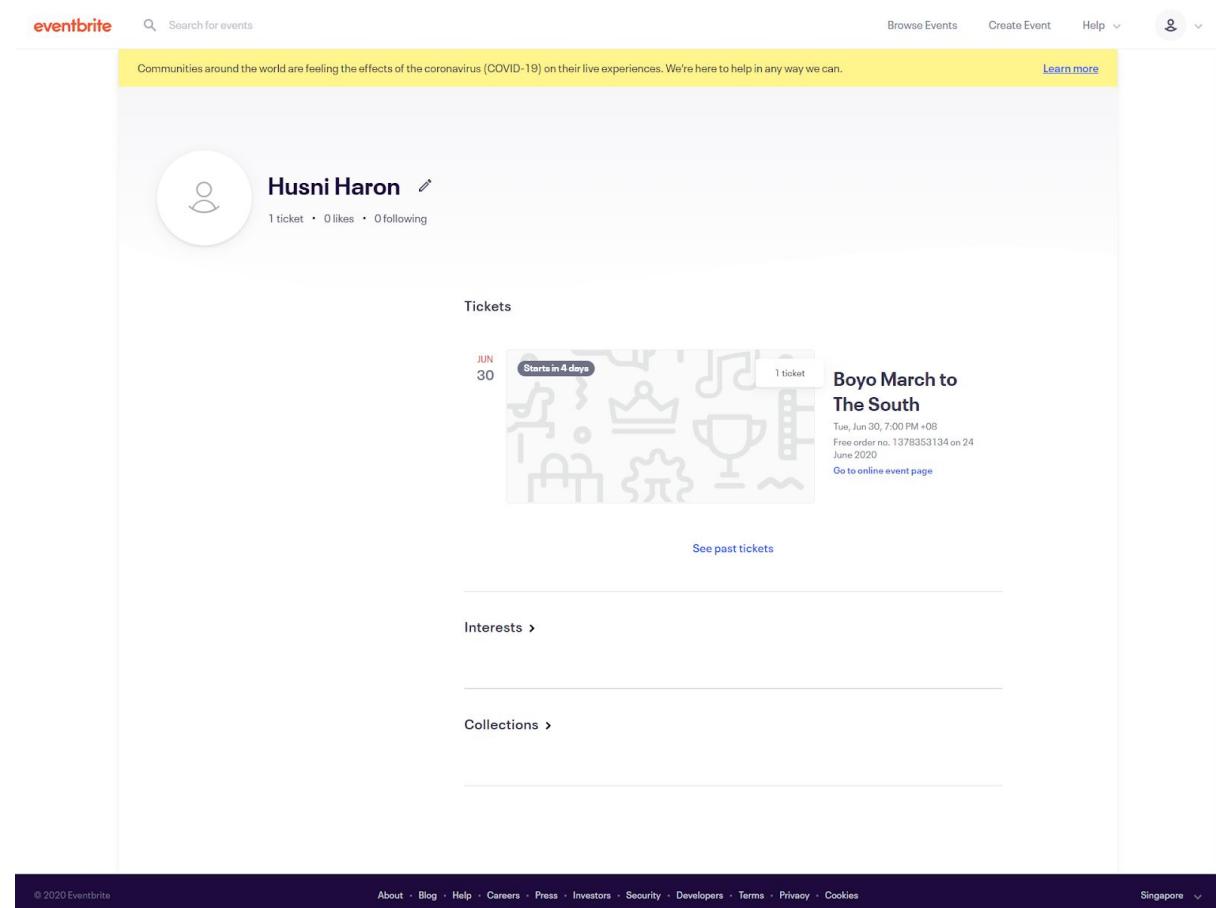
Diagram



Task 7

Eventbrite integrates with your private Telegram Bots API

Screenshots



eventbrite Search for events

Browse Events Create Event Help 

Tue, Jun 30, 2020 7:00 PM +08

Boyo March to The South

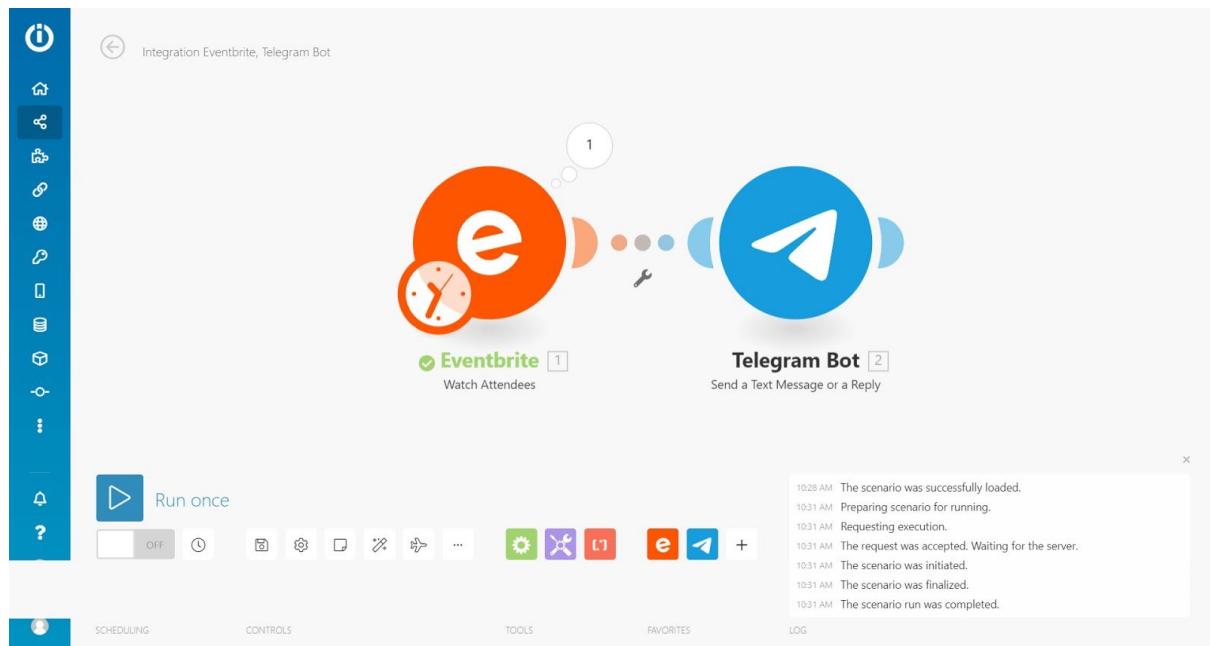
[View Event Details](#) 

 [Follow](#) [Contact](#)



Online Event Page not found
Links for this event are currently not available.

© 2020 Eventbrite [About](#) [Blog](#) [Help](#) [Careers](#) [Press](#) [Investors](#) [Security](#) [Developers](#) [Terms](#) [Privacy](#) [Cookies](#) United Kingdom 



eventbrite Search for events

Browse Events Create Event Help

JUN 30

Boyo March to The South

Free

Description
Me and the Boys march to the south..

Date And Time
Tue, Jun 30, 2020, 7:00 PM –
Wed, Jul 1, 2020, 10:00 PM SGT
[Add to Calendar](#)

Tags
[Online Events](#) [Online Rallies](#) [Online Seasonal Rallies](#)

Location
Online Event

Share With Friends

[Contact Event Organiser](#)

Other Events You May Like

 TUE, 17 NOV 21:00 FREE Payroll Webinar November 2020	 SUN, 28 MAR 03:00 School Project in Favela, Rio De Janeiro Brazil	 WED, 8 JUL 08:00 Allen and Gray Musical Festival - FULL PASS	 WED, 8 JUL 08:00 Allen and Gray Musical Festival - FULL PASS
#Business #Course	#FamilyEducation	#FilmMedia #Festival	#FilmMedia #Festival

Use Eventbrite
How It Works
Pricing

Plan Events
Online Registration
Sell Event Tickets
Event Management Software

Find Events
Browse Local Events
Get the Eventbrite App

Connect With Us
Report This Event
Help Centre
Terms
Privacy
Community Guidelines

© 2020 Eventbrite

Boyo March to The South
Tue, 30 Jun 2020 19:00 - Wed, 1 Jul 2020 22:00 +08

General Admission
Free
Sales end on 30 Jun 2020

Powered by **eventbrite** English (UK) ▾

Register

This screenshot shows the first step of a ticket purchase process. It displays a single ticket type, "General Admission", which is free and valid until June 30, 2020. The "Register" button is prominently displayed at the bottom.

Checkout
Time left 0:54

Contact Information

First name *	Surname *
Yoshi	Tosuru

Email *	Confirm email *
initial123d@gmail.com	initial123d@gmail.com

Keep me updated on the latest news, events, and exclusive offers from this event organizer.

Eventbrite can send me emails about the best events happening nearby.

By clicking "Register", I accept the [Terms of Service](#) and have read the [Privacy Policy](#). I agree that Eventbrite may [share my information](#) with the event organizer.

Powered by **eventbrite**

Register

This screenshot shows the second step of the process, "Checkout". It requires the user to provide contact information: first name (Yoshi), surname (Tosuru), email (initial123d@gmail.com), and a confirmation of the email. There are also two optional checkboxes for newsletter updates and local event notifications. The "Register" button is at the bottom.

Thanks for your order! #1379360554

YOU'RE GOING TO
Boyo March to The South

TICKET SENT TO
initial123d@gmail.com
[Change](#)

DATE
Tue, 30 Jun 2020 19:00 -
Wed, 1 Jul 2020 22:00 +08

View Tickets

This screenshot shows the final step of the process, "Thanks for your order!". It displays the order number (#1379360554), the event name ("Boyo March to The South"), the recipient's email ("initial123d@gmail.com"), and the event date and time. A "View Tickets" button is located at the bottom.

EventPeeps
2 members

June 25

WhoseKnee created the group «EventPeeps»

Helo 12:36 PM ✓✓
henlo 12:37 PM ✓✓
henlo 12:37 PM ✓✓
Henlo boyo 12:39 PM ✓✓
ifydegfw 12:40 PM ✓✓
Hi @eventpeeps_bot 12:42 PM ✓✓
Hello 12:45 PM ✓✓
ssfgjjksgjkdskjfd 12:45 PM ✓✓

WhoseKnee removed eventpeople
WhoseKnee added eventpeople

hhs 12:46 PM ✓✓

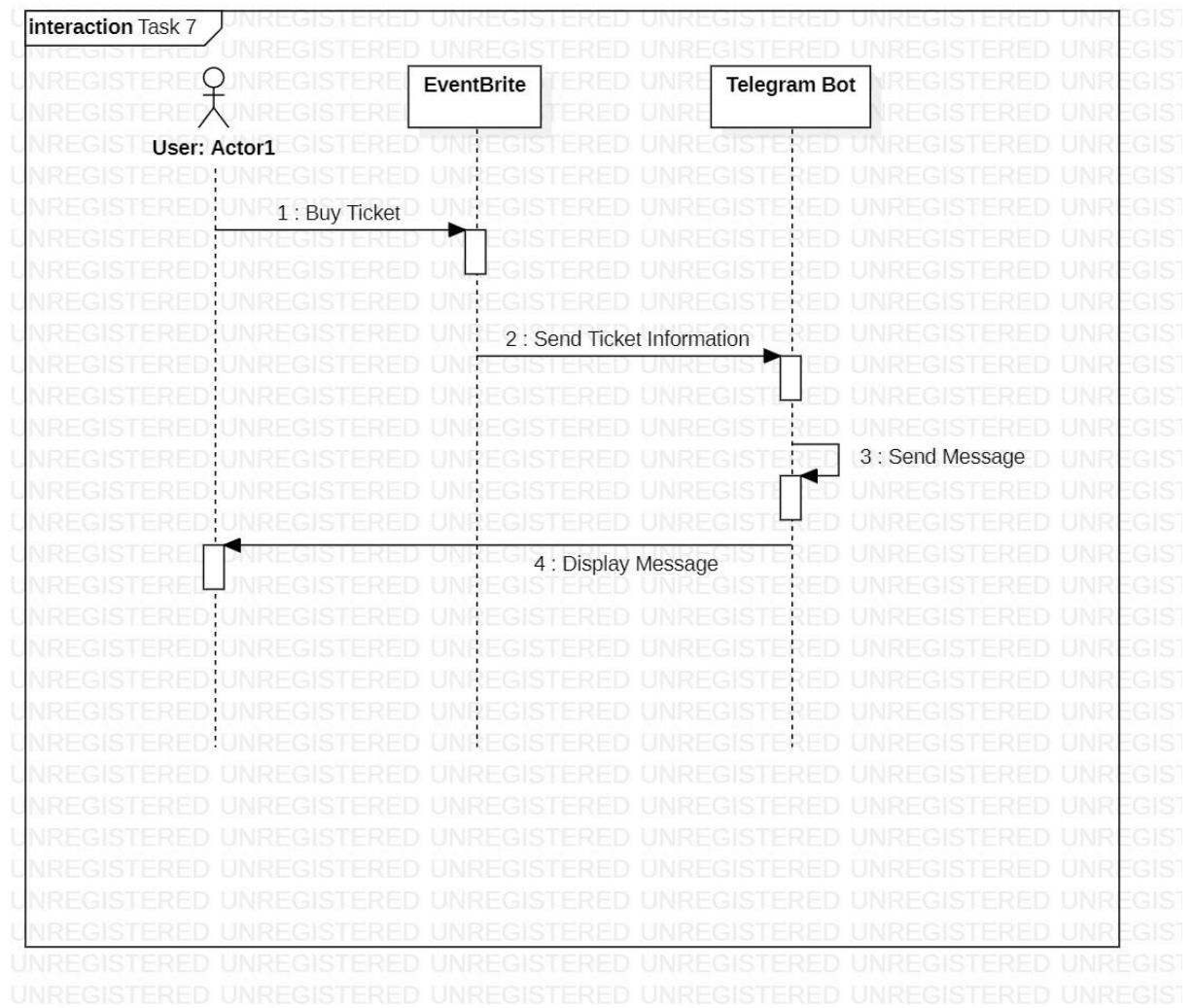
E eventpeople
Henlo, my boyos! 12:54 PM

June 26

eventpeople
Henlo, my boyos!
Here are the ticket details:
Yoshi Tosoru
189079162
General Admission 10:37 AM

Write a message...

Diagram



Task 8

Develop a web client that allows user to upload receipts images and tag the receipts image by calling web service from Clarifai API

Screenshots

Analyze My Image!

No file chosen

Analyze My Image!

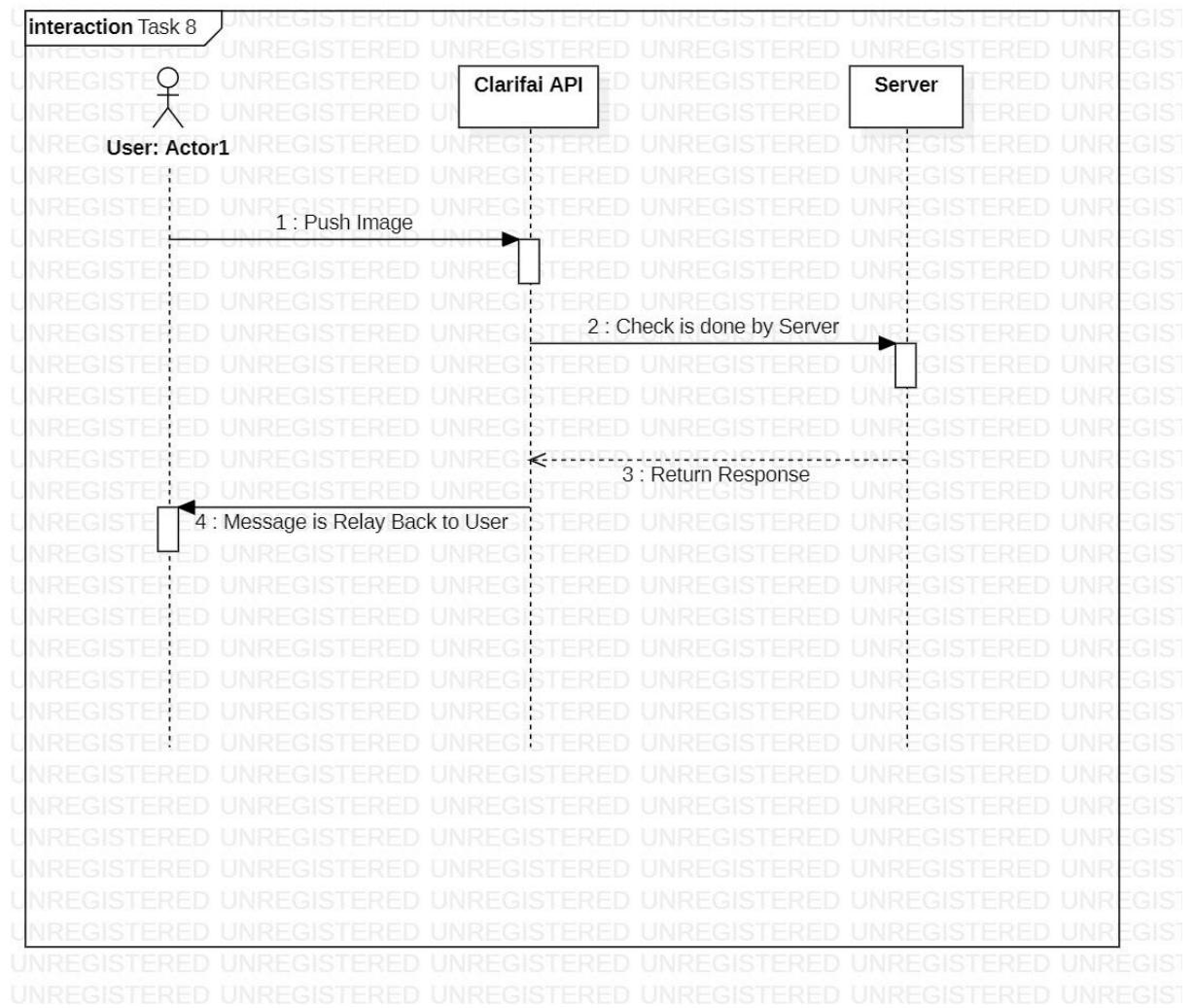
download (1).jfif



Food

The 3 food items we are most confident in detecting are:
- sauce at a 96.77% probability
- soup at a 96.16% probability
- vegetable at a 96.02% probability

Diagram

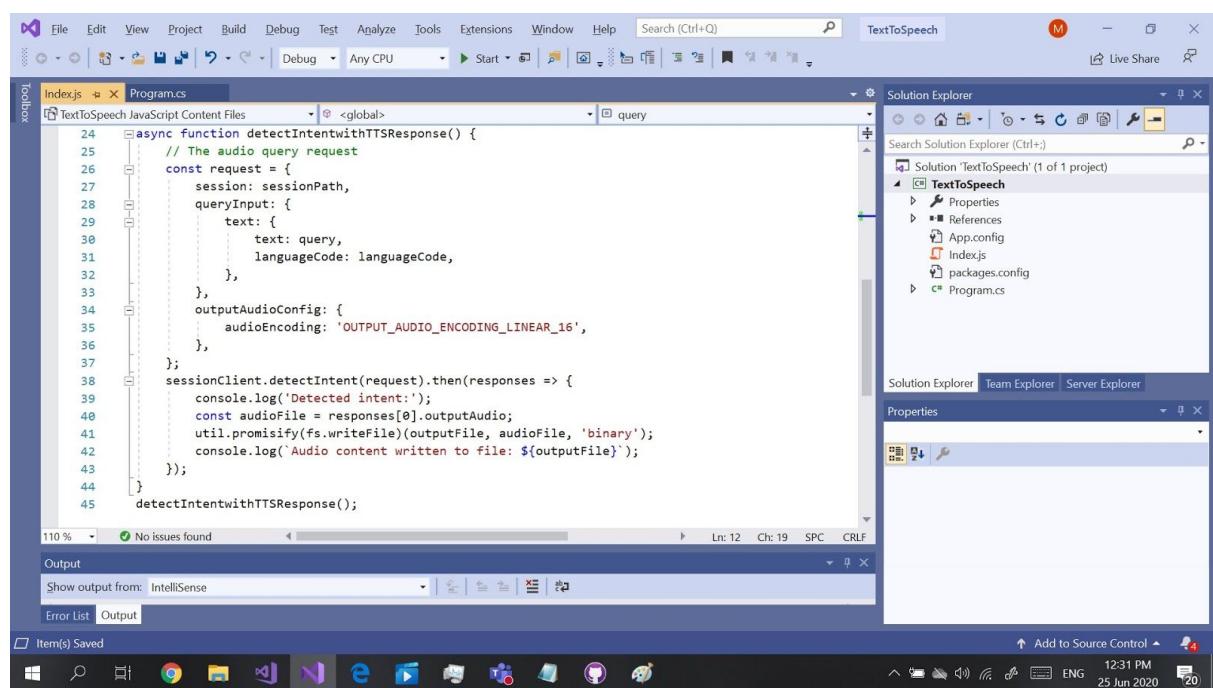
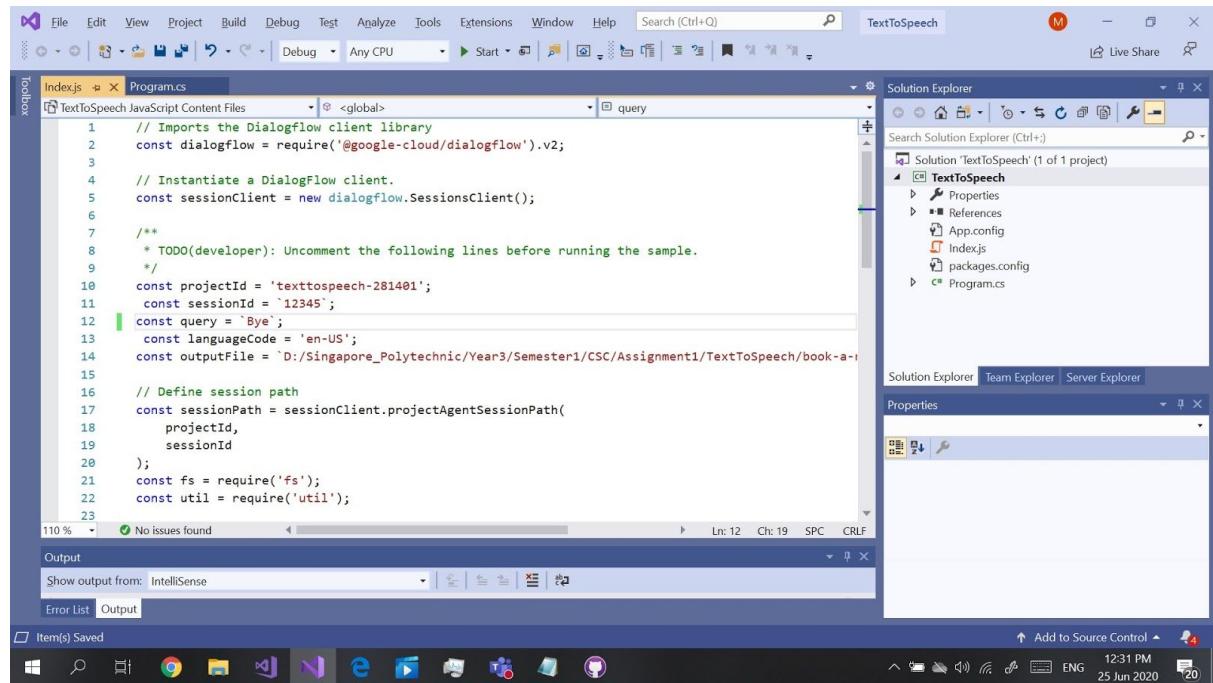


Extra Feature

Dialogflow API (Text-To-Speech)

Screenshots

Visual Studio



The screenshot shows the Microsoft Visual Studio interface. In the Solution Explorer, there is one project named 'TextToSpeech' containing files like App.config, Index.js, packages.config, and Program.cs. The Output window displays the command-line logs of the application's execution:

```

Index.js  Program.cs
Node.js command prompt
Your environment has been set up for using Node.js 12.18.1 (x64) and npm.
C:\Users\USER>cd C:\Users\USER\Source\Repos\TextToSpeech\TextToSpeech
C:\Users\USER\Source\Repos\TextToSpeech\TextToSpeech>set GOOGLE_APPLICATION_CREDENTIALS=C:\Users\USER\Source\Repos\TextToSpeech-a823aff7bf06.json
C:\Users\USER\Source\Repos\TextToSpeech\TextToSpeech>gcloud auth application-default print-access-token
ya29.c.Ko8B8AcG4jZTHUhdBwHsF0501zIiLxpCzzs-0g9QnUi0ezpcanydnugmEL8btRdCyQb7wOKUw2qdzYY2fjxEk0vVNLEnDX4amA4Idgnf_cTKeb
RNjhnnlxgcZ7Q0k8s4oBxuu419ovLfoSm0jxzI4uz5jHYRF5X4iO-v1zQx8sThqHmgFyk3yfxbaOLiYKhWY
C:\Users\USER\Source\Repos\TextToSpeech\TextToSpeech>node Index.js
Detected intent:
Audio content written to file: D:/Singapore_Polytechnic/Year3/Semester1/CSC/Assignment1/TextToSpeech/book-a-room.wav
C:\Users\USER\Source\Repos\TextToSpeech\TextToSpeech>

```

The screenshot shows the Google Dialogflow web interface. On the left, a sidebar menu includes options like TextToSpeech, en, Intent (which is selected and highlighted in blue), Entities, Knowledge (beta), Fulfillment, Integrations, Training, Validation, History, and Analytics. The main area is titled 'Intents' and contains a 'CREATE INTENT' button. Below it is a search bar and a list of intents: Default Fallback Intent, Default Welcome Intent, and room.reservation. To the right, there is a 'Try it now' section with a 'See how it works in Google Assistant' button, an 'Agent' section showing a user saying 'good day' and an AI response 'Hi! I'm your conference room booking bot. I can help you to find a perfect room for your meeting and manage your reservations. What would you like to do?', and an 'INTENT' section showing 'Default Welcome Intent'. The status bar at the bottom indicates the time as 11:02 AM and the date as 25 Jun 2020.

The screenshot shows the Google Cloud Platform API & Services Credentials page. The left sidebar is titled "API" and lists "APIs & Services". Under "Credentials", there are three sections: "API keys", "OAuth 2.0 Client IDs", and "Service Accounts".

API keys:

Name	Creation date	Restrictions	Key	Usage with all services (last 30 days)
No API keys to display				

OAuth 2.0 Client IDs:

Name	Creation date	Type	Client ID
⚠		Service account client	11363545899870410...

Service Accounts:

Email	Name	Usage with all services (last 30 days)
dialogflow-npuovy@texttospeech-281401.iam.gserviceaccount.com	Dialogflow Integrations	20
texttospeech@texttospeech-281401.iam.gserviceaccount.com	TextToSpeech	14

At the bottom, there is a Windows taskbar with various icons and system status information.