API Documentation

Bukidnon State University
Malaybalay City

College of Technologies
Information Technology Department
Bachelor of Science in Information Technology
1st Semester SY: 2022 – 2023
IT137 – Integrative Programming and Technologies 2

SBO Fee Collection Management System
API Documentation

I. Overview

1.1 Project Name
SBO Fee Collection Management System

1.2 Client / Respondents of the System
Louie Jay S. Labastida, COT-SBO Adviser

1.3 Description
The Student Body Organization (SBO) Fee Collection Management System is a comprehensive, web-based platform designed to help officers of the College of Technology Student Body Organization (COT SBO) manage their fees. This all-inclusive solution strives to reduce human error while offering a user-friendly interface for managing student fees and their categories of payments. The system guarantees accurate recording and arrangement of financial data pertaining to COT officers daily dues by providing an intuitive interface. The system's capacity to create and email receipts to students is a crucial feature that improves fee collection transparency by giving students instant proof of successful payments.

The system incorporates distinct user roles with specific permissions to maintain data integrity and security. Treasurers are authorized to enter daily dues and collect payments, while officers are granted access to view the payments and generate reports. In addition, the COT-SBO governor manages students in the system while the administrator has the responsibility of adding and archiving SBO officers. This role-based structure ensures that each user can perform their designated tasks effectively while maintaining appropriate access controls. The automated email receipt generation further streamlines the workflow, reducing manual administrative tasks and providing students with instant proof of payment. It also generates comprehensive reports on fee collection for the organization.

The SBO Fee Management System is designed to improve the overall efficiency of the organization's financial activities. It offers a comprehensive solution for handling student fees, classifying various sorts of payments, keeping an up-to-date database of COT officers, and communicating payment status clearly via automated receipts. By automating several areas of fee administration, the system considerably minimizes the possibility of human error, which improves financial record accuracy and increases student satisfaction. This specialized approach to charge

management allows the COT-SBO to focus on better servicing its officers while also ensuring a dependable, user-friendly financial tracking system that keeps every member informed and up to date.

1.4 Key Features:

- **Secure Login**: Utilizes Google Authentication for safe access, ensuring that only authorized users can log in.

- **Error Handling:** Implements error handling for invalid login attempts and unauthorized actions to enhance security.

- **Student Management:** Allows COT-SBO Officers, Treasurers, and Governors to add and manage student information seamlessly.

- **Payment Reporting:** Generates comprehensive reports on student payments, including monthly summaries and categorized data for efficient tracking.

- **Receipt Management:** Facilitates viewing and generating receipts for payments, with email notifications for transparency.

- **Google Calendar:** Integrates a calendar feature for scheduling payment deadlines, student meetings, and important reminders, ensuring timely follow-ups and payment tracking.

- **Email (For sending payment details):** Allows automated email notifications to students and relevant parties with payment details, receipt confirmations, and payment reminders, enhancing communication and transparency.

- **Google Drive:** Provides integration with Google Drive for storing and sharing payment-related documents, reports, and other important files securely, ensuring easy access for authorized users.

- **Role-Based Access:** Restricts system access based on user roles to maintain data integrity and security.

- **Archive, Unarchive Student Records, and Manage Accounts:** Archive and unarchive student records while disabling accounts upon resignation to maintain an organized system.

- **Secure Logout:** Ensures a secure logout process to protect user information after sessions.

1.5 Version

1.0.0 – Beta

1.6 Base URL

http://localhost:8000/

1.7 Authentication

The SBO Fee Collection Management System uses JWT (JSON Web Token) for secure user authentication. To prevent automated attacks, Google reCAPTCHA is integrated into the login process, requiring users to verify they are human. Once the reCAPTCHA is validated, the user's credentials are checked, and if valid, a JWT token is issued for the session.

II. Endpoints

The following are the detailed endpoints of SBO Fee Collection Management System API. These include HTTP methods used in the API call, parameters, configuration, request and response of the API.

2.1 AUTHENTICATION AND GENERAL MODULE API

2.1.1 Module Description

The COT-SBO system provides endpoints to manage user authentication, session control, and secure operations. These include endpoints for user login, Google user verification, profile retrieval and updates, OTP generation and verification, and session termination. Each endpoint ensures secure access through token-based authentication, validates user input, and protects data integrity. The API employs comprehensive response codes: 200 (OK) for successful operations, 201 (Created) for resource creation, 400 (Bad Request) for invalid input, 401 (Unauthorized) for missing or invalid tokens, 403 (Forbidden) for insufficient permissions, 404 (Not Found) for unavailable resources, and 500 (Internal Server Error) for unexpected server issues, ensuring secure and clear communication of request outcomes.

The API have the following endpoints:

http://localhost:8000/api/login

http://localhost:8000/api/auth/verify-google-users

http://localhost:8000/api/profile.1/:email/:position

http://localhost:8000/api/profile/:email/:position

http://localhost:8000/api/send-otp

http://localhost:8000/api/verify-otp

http://localhost:8000/api/logout

Response codes of this API:

Code	Message	Description
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
429	Too Many Request	The client sends too many requests within a certain period.
404	Not Found	The requested resource could not be found.
403	Restricted	Invalid token or token expired.
401	Unauthorized	The request was not successful because it lacks valid authentication credentials.
400	Bad Request	The request was invalid.
201	Created	The request was successful, and a new resource has been created.
200	OK	The request is succeeded, the resource is in the message body.

2.1.1.1 Authenticate User and Create Session

Version: 1.0

Date: October 22, 2024

Description: This API endpoint enables the creation of login credentials for secure access to the College of Technology Student Body Organization (COT-SBO) system. Error responses are generated for invalid login attempts, including incorrect credentials or missing parameters. This ensures that only registered officers with valid login details can access the COT-SBO system.

Endpoint:

```
http://localhost:8000/api/login
```

Method: POST

Configurations:

The API request must be authenticated to ensure secure access to the COT-SBO login creation process. The request must include a valid token in the header to verify the identity of the user making the request.

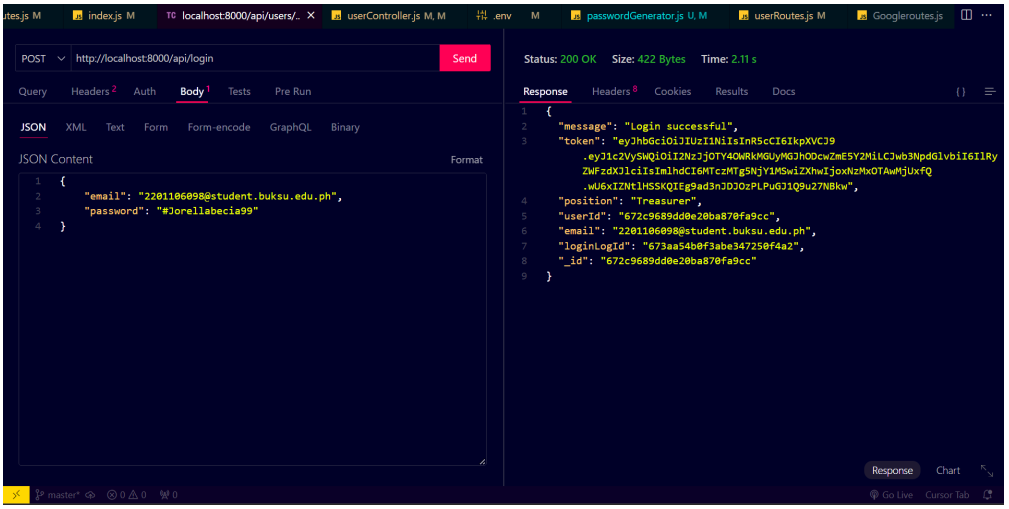
Parameters:

- **_token -** this is required for handling requests to this API endpoint. The token should be sent in the request header to verify the identity of the user making the request.

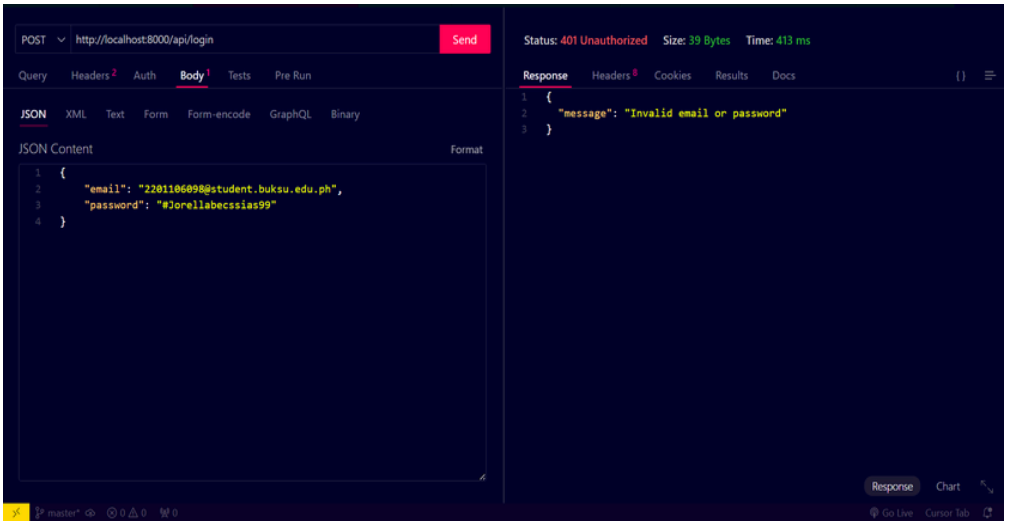
- **_username** – this is required to specify the user's username. It should be included in the request body.
- **_password** – this is required for the user's password. It should be included in the request body to ensure secure login credentials.
- **_role** – this is required to define the user's role within the organization. It should also be included in the request body.
- **_recaptchaResponse** – this is required to verify the reCAPTCHA response token submitted by the officer. It should be included in the request body for verification.
- **_rememberMe** – this is optional and indicates whether the login session should be remembered for future access. It can be included in the request body.
- **_errorMessage** – this is optional and is returned in the response body to provide details about any login issues, such as "Invalid username or password" or "reCAPTCHA verification failed."

Requests:

Valid Request



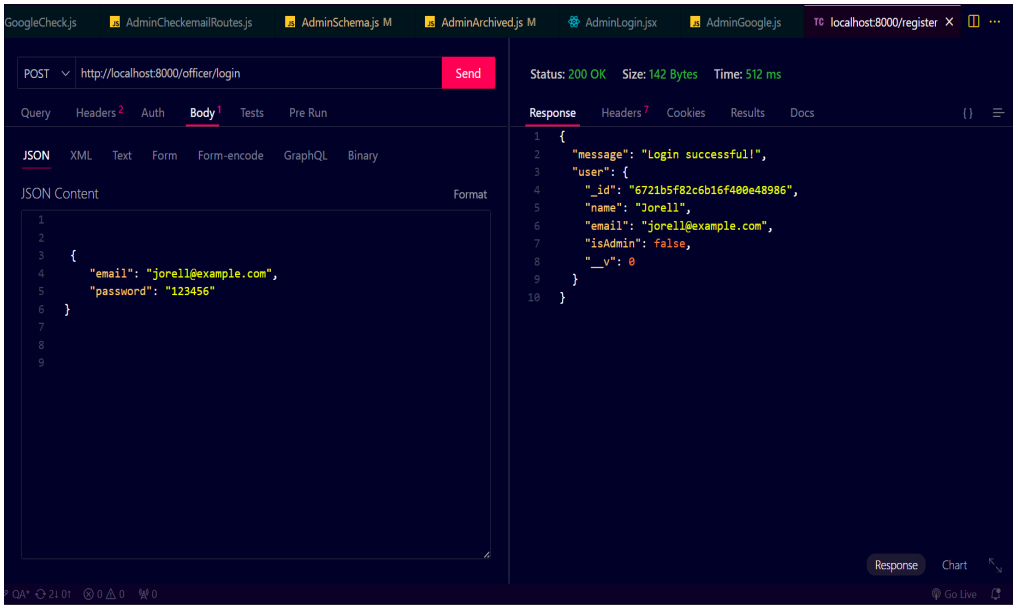
Not Valid Request

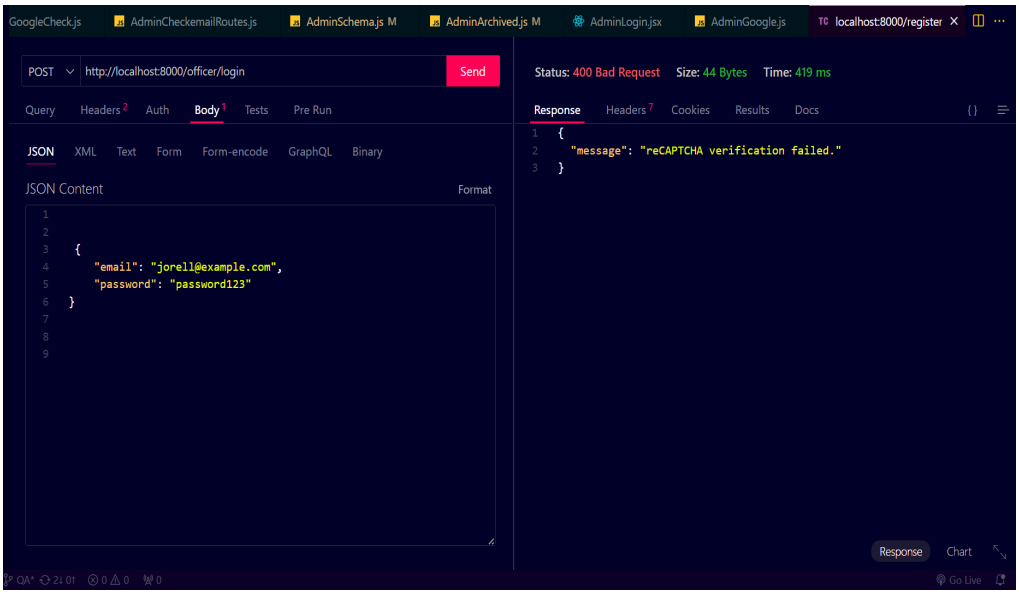


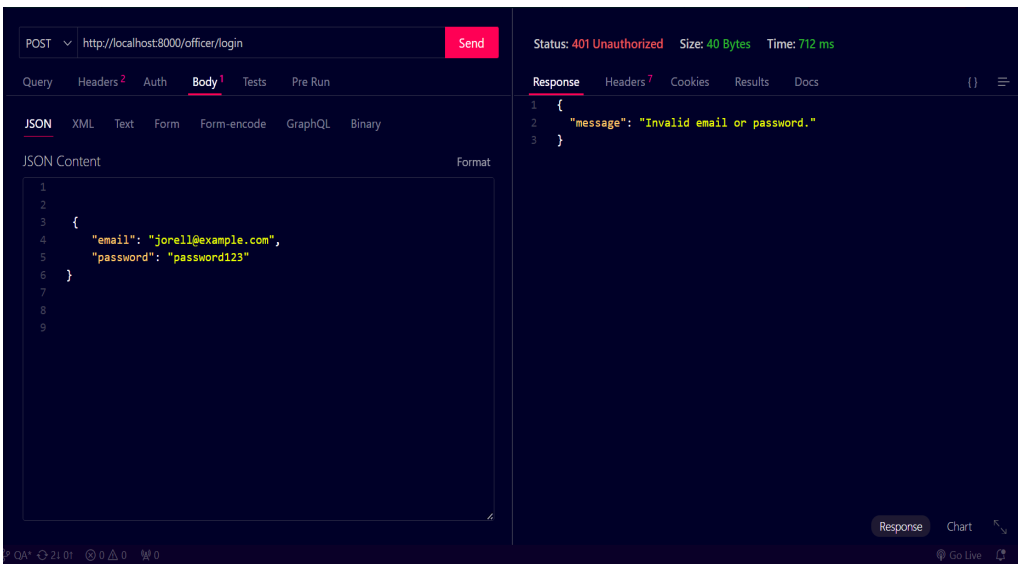
Response Format: JSON

Responses:

Success Response







The success response for this API includes a 200 OK status code and a success message confirming the creation of the user's login credentials, allowing access to the COT-SBO system. The response contains the user's username, email, and ID within the organization. While failed reCAPTCHA responses are excluded in this case due to separate handling of reCAPTCHA functionality, such scenarios previously returned a 400 Bad Request status code with the message "reCAPTCHA verification failed." Additionally, unauthorized access due to a missing or invalid authentication token triggers a 401 Unauthorized status code with the message "Invalid email or password," ensuring that secure access policies are enforced.

2.1.1.2 Verify Google User Authentication

Version: 1.0

Date: October 22, 2024

Description: This API endpoint verifies the credentials of users attempting to log in through Google authentication within the College of Technology Student Body Organization (COT SBO) system. It helps to ensure that the user is authenticated via Google and securely grants access to the system. This feature is essential for providing an efficient authentication process, enhancing user experience, and maintaining the system security.

Endpoint:

```
http://localhost:8000/api/auth/verify–google-users
```

Method: GET

Configurations:

The API request must be authenticated to ensure secure access to user data. Proper input validation should be conducted to ensure the Google token is valid and that the required parameters (email, google_token) are included. Additionally, the API may implement rate limiting to protect against excessive requests and ensure best system performance.

Parameters:

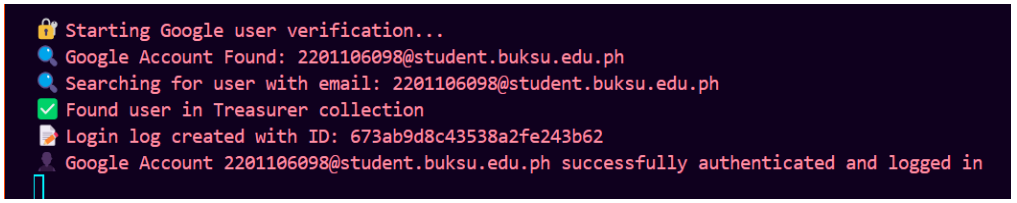
- **_token** – This is required for handling requests to this API endpoint. The token should be sent in the request header to verify the identity of the user making the request.
- **google_token** – This is required. The token received from Google authentication to verify the user's identity and access permissions.
- **email** – This is required to specify the Google email address of the user attempting to log in.

➢ **device_id** (optional) – This parameter can be used to identify the device from which the request is being made, providing additional security features.

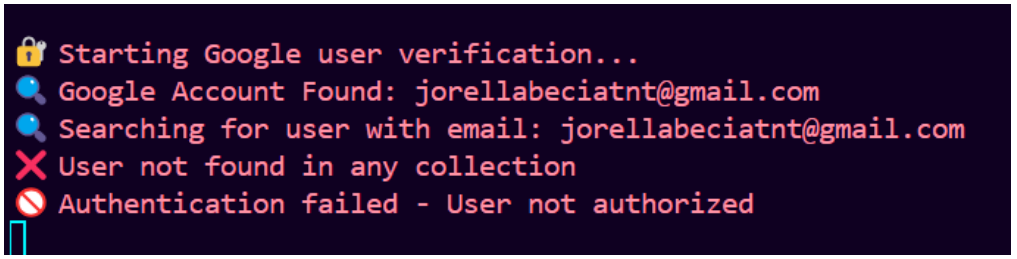
➢ **redirect_url** – This is an optional parameter which specifies the URL to redirect the user to upon successful authentication.

Requests:

Valid Request



Not Valid Request



Response Format: Console Log

Response:

The API response confirms the successful verification of the user via Google authentication within the College of Technology Student Body Organization (COT SBO) system. If the provided Google token is valid and the request is correctly authenticated, the response will return an authentication token that grants secure access to the system, along with a success message. If the verification fails (e.g., invalid token, incorrect email, or unauthorized access), the response will provide an error message, such as "Invalid Google token," "Authentication failed," or "Unauthorized access attempt," guiding the user to resolve the issue and ensuring that only legitimate users can access the system.

2.1.1.3 Retrieve Profile by Email and Position

Version: 1.0

Date: October 22, 2024

Description: This API endpoint enables the retrieval of a user's profile based on their email address and position within the College of Technology Student Body Organization (COT-SBO) system. It ensures that only authenticated users can access sensitive profile information. Invalid or unauthorized access attempts will generate appropriate error responses.

Endpoint:

```
http://localhost:8000/api/profile/:email/:position
```

Method: GET

Configurations:

The API request must be authenticated to ensure secure access to user profiles. The request must include a valid token in the header to verify the identity of the user making the request.

Parameters:

- **_token** - this is required and must be included in the request header for user authentication.
- **_email** - this is required and specifies the email address of the user whose profile is being retrieved. It should be passed as a path parameter.
- **_position** - this is required and defines the position of the user (e.g., "student," "officer," etc.). It should also be passed as a path parameter.
- **_errorMessage** - this is optional and will be returned in the response body if the request fails, providing details such as "Invalid token" or "Profile not found."

Requests:

Valid Request


```
{
    "request": {
        "method": "GET",
        "url": "http://localhost:8000/api/profile/john.doe@schooldistrict.gov/finance_official"
        "headers": {
            "Content-Type": "json",
            "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
        }
    },
```


Not Valid Request


```
    {
      "status": 400,
      "contentType": "json",
      "body": {
        "error": "Invalid Input",
        "details": "Invalid email format or missing position parameter"
      }
    },
```


Response Format: JSON

Responses:

Successful Response


```
    "response": {
        "status": 200,
        "statusText": "OK",
        "headers": {
            "Content-Type": "json",
            "X-Request-ID": "req-2024-02-15-123456"
        },
        "body": {
            "personalInfo": {
                "firstName": "John",
                "lastName": "Doe",
                "email": "john.doe@schooldistrict.gov",
                "contactNumber": "+1-555-987-6543",
                "profilePicture": "https://example.com/profiles/john-doe.jpg",
                "dateOfBirth": "1985-04-15",
                "gender": "Male"
            },
            "professionalDetails": {
                "position": "finance_official",
                "department": "Financial Services",
                "employeeId": "SBO-2023-0045",
                "specialization": "Fee Collection Management",
                "employmentType": "Full-time",
                "hireDate": "2020-01-15"
            },
```


Error Response


```
{
    "request": {
        "method": "GET",
        "url": "http://localhost:8000/api/profile/john.doe@schooldistrict.gov
        "headers": {
            "Content-Type": "json",
            "Authorization": "Bearer invalidToken123"
        }
    },
    "response": {
        "status": 401,
        "statusText": "Unauthorized",
        "headers": {
            "Content-Type": "json",
            "X-Request-ID": "req-2024-02-15-345678"
        },
```

```
  "response": {
    "status": 403,
    "statusText": "Forbidden",
    "body": {
      "success": false,
      "message": "Access to admin profile denied",
      "error": "Insufficient privileges",
      "errorCode": "PROFILE_ACCESS_DENIED"
    }
  }
}
```

```
    "response": {
        "status": 404,
        "statusText": "Not Found",
        "headers": {
            "Content-Type": "json",
            "X-Request-ID": "req-2024-02-15-234567"
        },
        "body": {
            "error": "Profile Not Found",
            "message": "No profile exists for the specified email and position",
            "errorCode": "PROFILE_NOT_FOUND",
            "details": {
                "email": "nonexistent@schooldistrict.gov",
                "position": "admin"
            }
        }
    }
}
```

```
  "response": {
    "status": 404,
    "statusText": "Not Found",
    "body": {
      "success": false,
      "message": "Admin profile not found",
      "error": "No profile associated with the current authentication token",
      "errorCode": "PROFILE_NOT_FOUND"
    }
  }
}
```

```
  "response": {
    "status": 500,
    "statusText": "Internal Server Error",
    "body": {
      "success": false,
      "message": "Unable to retrieve admin profile",
      "error": "Database connection error",
      "errorCode": "PROFILE_RETRIEVAL_FAILED",
      "timestamp": "2024-02-15T10:55:33.221Z"
    }
  }
}
```

The API returns a 200 OK status for successfully retrieving a user profile based on email and position, confirming the operation's success. A 401 Unauthorized status indicates missing or invalid authentication tokens, ensuring only authorized users can access profile data. A 403 Forbidden status signifies that the user lacks sufficient permissions for the request, while a 404 Not Found is returned if the specified profile does not exist. For unexpected server issues, a 500 Internal Server Error advises users to retry or seek support. These responses ensure secure, clear, and efficient communication about the request's outcome.