

API Documentation (V1.3)
for
Higher Education Management Information System
(HEMIS)

University Grants Commission (UGC)

www.ugcnepal.edu.np

Prepared by:

Debugsoft Pvt. Ltd.

dibugsoft@gmail.com

www.dibugsoft.com

January 2025

Contents

Contents	2
1. Introduction.....	4
2. Purpose.....	4
3. Key Features	4
4. Audience	4
Getting Started.....	5
5. API Documentation (V1.3).....	5
5.1. API Documentation for the Infrastructure Management Module	6
5.1.1. API Documentation – Buildings	6
5.1.2. API Documentation: Furniture, Equipment, Vehicle.....	9
5.1.3. API Documentation: Hostels Management	12
5.1.4. API Documentation: Lab Management	17
5.1.5. API Documentation: Land Management.....	22
5.1.6. API Documentation: Library Management	26
5.2. API documentation for BATCH	33
5.3. API documentation for Fiscal Year	34
5.4. API documentation to get Program.....	34
5.5. API to get address for student register, employee register.....	35
5.6. Register/Add Department and Section for campus	36
5.8. API Documentation for Student Management	40
5.8.1. Introduction.....	40
5.8.2. Register Student	40
5.8.3. Get All Students	46
5.8.4. Get Student by ID	48
5.8.5. Update Student.....	49
5.9. API for student Upgrade	53
5.9.1. Introduction.....	53
5.9.2. Upgrade Students	53
5.10. API Document for Employee Position.....	57
5.10.1. Introduction.....	57
5.11. API Documentation: Employee Management	58
5.11.1. Introduction.....	58
5.11.2. Get Employee.....	58

- 5.11.3. **POST Employee** 61
- 5.11.4. Update Employee** 64
- 5.12. API for Graduate Student..... 68
- 5.12.1. Introduction**..... 68
- 5.13. API for Dropout Student 73
- 5.13.1. Introduction**..... 73

Implementation of API for Higher Education Management System (HEMIS)

1. Introduction

Higher Education Management Information System (HEMIS) API is designed to facilitate seamless integration and management of academic, administrative, and operational processes in higher education institutions. It enables universities, colleges, and other educational organizations to automate and manage student records, faculty details, enrollment processes, document storage, and various administrative functions with a centralized database.

2. Purpose

The **HEMIS API** provides a comprehensive suite of endpoints to handle key aspects of an institution's operations, including:

- **Student Management:** Enrollment, academic records, personal details, and certifications.
- **Student Management:** Registration and management the student enrollment.
- **Student Upgrade:** Upgrade student from current semester/year to next.
- **Graduation:** Recording the graduation details of each student.
- **Employee Management:** Profiles, positions detail.

3. Key Features

- **Secure & Scalable:** Built with robust authentication mechanisms ensuring data security and compliance with educational standards.
- **Comprehensive Data Handling:** Manage detailed records for students, faculty, and institutional processes.
- **Integration Ready:** Easily connect with existing educational platforms, ERP systems, and government databases.
- **Flexible & Customizable:** Designed to accommodate various higher education structures and policies.

4. Audience

This documentation is intended for:

- Developers integrating **HEMIS API** into university and campus/college management systems.
- IT teams managing student records and faculty databases.
- Government agencies overseeing higher education compliance and reporting.

Getting Started

To begin using the **HEMIS API**, follow these steps:

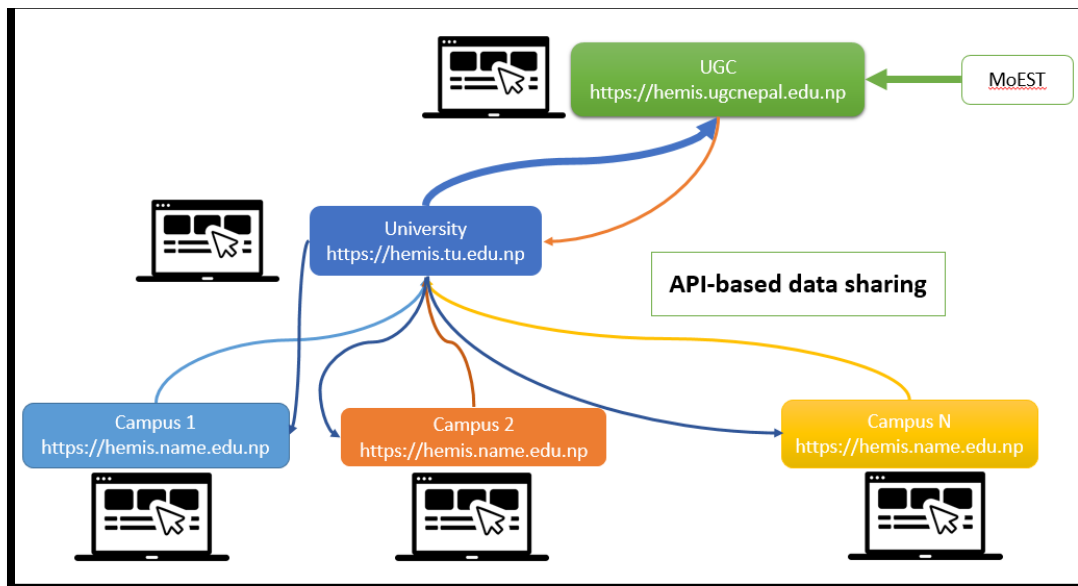
1. **Authentication:** Obtain API credentials to access secure endpoints.
2. **Explore Endpoints:** Review available endpoints to manage student, faculty, and institutional data.
3. **Test & Implement:** Use sample requests and responses to integrate the API into your system.

5. API Documentation (V1.3)

This is an API document to integrate campus-level HEMIS with the University and UGC HEMIS.

There are various APIs to GET, POST, PUT/PATCH data from/to a centralized, integrated database as required.

Before integrating/using the API user should understand the following graph:



Data flow diagram of the centralized, integrated system.

The key setting parameters managed by a higher level should be obtained as these are in the centralized database via GET API from the higher level and the data produced in the campuses should be shared via POST API to the higher level. The whitelisted IP is only authorized to POST, PUT/PATCH, and GET data in PRODUCTION MODE. However, any IP is allowed for deployment mode to test the system.

User should use the base URL (<https://hemisapi.ugcnepal.edu.np>) before `/api/` in this document.

To get the Token for this test API: Request a POST in Postman:

POST: <https://hemisapi.ugcnepal.edu.np/api/User/Login>

Pass the following parameter in JSON format in the body:

```
{
  "email": "userexample@gmail.com",
  "password": "password",
  "collegeId": collegeId,
  "UniId": 0,
  "isUgc": false,
}
```

You will get the TOKEN in the response body.

5.1. API Documentation for the Infrastructure Management Module

This API helps to POST, GET, PUSH/PATCH all the required information of infrastructure as per UGC guidelines. The actual data/information related to Land, Building, Lab, Hostel, Facility, Library, and Equipment available on the particular campus should be tracked and reported in real-time to UGC via this API. This data supports getting infrastructure for regular grants and other purposes to UGC.

5.1.1. API Documentation – Buildings

`/api/Buildings` - to register all buildings in the system. Users can use `/api/Buildings/{buildingId}` to get particular building data.

GET `/api/Buildings/{buildingId}`

- **Tags:** Buildings
- **Description:** Retrieve all building records.
- **Headers:** Authorization: Bearer

- **Responses:**
 - 200: Success (Returns an array of BuildingsDto)

```
{
  "id": 3,
  "houseName": "Block C",
  "areaCoveredByBuilding": 2500,
  "noOfClassrooms": 10,
  "areaCoveredByAllRooms": 200,
  "ownershipOfBuilding": true,
  "hasInternetConnection": true,
  "remarks": "test"
}
```

POST

- **Tags:** Buildings
- **Description:** Create a new building entry.
- **Headers:** Authorization: Bearer
-
- **Request Body:** application/json

```
{
  "id": 0,
  "houseName": "string",
  "areaCoveredByBuilding": 0,
  "noOfClassrooms": 0,
  "areaCoveredByAllRooms": 0,
  "ownershipOfBuilding": true,
  "hasInternetConnection": true,
  "remarks": "string"
}
```

- **Responses:**

- 200: Success (Returns created BuildingsDto)

```
{
  "id": 0,
  "houseName": "string",
  "areaCoveredByBuilding": 0,
  "noOfClassrooms": 0,
  "areaCoveredByAllRooms": 0,
  "ownershipOfBuilding": true,
  "hasInternetConnection": true,
  "remarks": "string"
}
```

PUT: to get all buildings registered in the system. Users can use `/api/Buildings/{buildingId}` to get particular building data.

- **Tags:** Buildings
- **Description:** Update an existing building entry.
- **Parameters:**
 - id (integer, int32, required, in path)
- **Request Body:** application/json

```
{
  "id": 3,
  "houseName": "Block C",
  "areaCoveredByBuilding": 2500,
  "noOfClassrooms": 10,
  "areaCoveredByAllRooms": 200,
  "ownershipOfBuilding": true,
  "hasInternetConnection": true,
  "remarks": "test"
}
```

Responses:

```
{
```

```
"id": 3,  
"houseName": "Block C",  
"areaCoveredByBuilding": 2500,  
"noOfClassrooms": 10,  
"areaCoveredByAllRooms": 200,  
"ownershipOfBuilding": true,  
"hasInternetConnection": true,  
"remarks": "test"  
}
```

- 200: Success (Returns updated BuildingsDto)

5.1.2. API Documentation: Furniture, Equipment, Vehicle

This API help to POST, PUT and GET equipment, furniture, vehicle detail to centralized server and vice versa.

Endpoints:

- GET /api/FurnitureEquipmentVehicle
- POST /api/FurnitureEquipmentVehicle
- PUT /api/FurnitureEquipmentVehicle/{id}

Description:

These endpoints allow retrieving, adding, and updating furniture, equipment, and vehicles in the system.

GET /api/FurnitureEquipmentVehicle

Request

- **Headers:** Authorization: `Bearer {Token}`
- **Query Parameters:** None.
- **Request Body:** Not applicable.

Response (200)

Description: Returns a list of furniture, equipment, and vehicles.

Content-Type: application/json, text/json, text/plain

Response Body:

```
[
  {
    "id": 0,
    "itemType": "string",
    "itemName": "string",
    "noOfQty": 0,
    "itemDescription": "string",
    "remarks": "string"
  }
]
```

POST /api/FurnitureEquipmentVehicle*Request*

- **Headers:** Content-Type: application/json, Authorization: `Bearer {Token}`

Request Body:

```
{
  "itemType": "string",
  "itemName": "string",
  "noOfQty": 0,
  "itemDescription": "string",
  "remarks": "string"
}
```

Response (201)

Description: Furniture, equipment, or vehicle item added successfully.

Response Body:

```
{
  "id": 0,
  "itemType": "string",
  "itemName": "string",
```

```
"noOfQty": 0,  
"itemDescription": "string",  
"remarks": "string"  
}
```

PUT /api/FurnitureEquipmentVehicle/{id}

Request

Headers: Content-Type: application/json

Path Parameter:

id (integer, required) - The ID of the item to be updated.

Request Body:

```
{  
  "id":0,  
  "itemType": "string",  
  "itemName": "string",  
  "noOfQty": 0,  
  "itemDescription": "string",  
  "remarks": "string"  
}
```

Response (200)

Description: Item updated successfully.

Response Body:

```
{  
  {  
    "id":0,  
    "itemType": "string",  
    "itemName": "string",  
    "noOfQty": 0,  
    "itemDescription": "string",  
    "remarks": "string"  
  }  
}
```

Error Responses

400 Bad Request

Cause: Malformed request or invalid data.

Example Response:

```
{  
  "error": "Invalid request format."  
}
```

404 Not Found

Cause: Requested item does not exist.

Example Response:

```
{  
  "error": "Item not found."  
}
```

500 Internal Server Error

Cause: Unexpected server issue.

Example Response:

```
{  
  "error": "An unexpected error occurred. Please try again later."  
}
```

Authorization

No authentication or authorization is required for these endpoints.

Notes

- These endpoints facilitate managing furniture, equipment, and vehicles.
- Ensure that data integrity is maintained when updating or adding items.

5.1.3. API Documentation: Hostels Management

Endpoints:

- GET /api/Hostels
- POST /api/Hostels
- GET /api/Hostels/{id}

- PUT /api/Hostels/{id}

Description:

These endpoints allow retrieving, adding, and updating hostel details in the system.

GET /api/Hostels

Request

- **Headers:** Authorization: `Bearer {Token}`
- **Query Parameters:** None.
- **Request Body:** Not applicable.

Response (200)

Description: Returns a list of hostels.

Content-Type: application/json, text/json, text/plain

Response Body:

```
[
  {
    "id": 0,
    "hostelType": "string",
    "noOfRoomsInHostel": 0,
    "noOfSeats": 0,
    "areaCoveredByHostel": 0,
    "buildingId": 0,
    "hasPlayground": true,
    "hasInternet": true,
    "hasDrinkingWater": true,
    "hasToilet": true,
    "remarks": "string"
  }
]
```

POST /api/Hostels

Request

Headers: Content-Type: application/json, Authorization: `Bearer {Token}`

Request Body:

```
{
  "hostelType": "string",
  "noOfRoomsInHostel": 0,
  "noOfSeats": 0,
  "areaCoveredByHostel": 0,
  "buildingId": 0,
  "hasPlayground": true,
  "hasInternet": true,
  "hasDrinkingWater": true,
  "hasToilet": true,
  "remarks": "string"
}
```

Response (201)

Description: New hostel added successfully.

Response Body:

```
{
  "id": 0,
  "hostelType": "string",
  "noOfRoomsInHostel": 0,
  "noOfSeats": 0,
  "areaCoveredByHostel": 0,
  "buildingId": 0,
  "hasPlayground": true,
  "hasInternet": true,
  "hasDrinkingWater": true,
  "hasToilet": true,
  "remarks": "string"
}
```

GET /api/Hostels/{id}

Request

- **Headers:** Authorization: `Bearer {Token}`.
- **Path Parameter:**
 - id (integer, required) - The ID of the hostel to retrieve.

Response (200)

Description: Returns hostel details.

Response Body:

```
{
  "id": 0,
  "hostelType": "string",
  "noOfRoomsInHostel": 0,
  "noOfSeats": 0,
  "areaCoveredByHostel": 0,
  "buildingId": 0,
  "hasPlayground": true,
  "hasInternet": true,
  "hasDrinkingWater": true,
  "hasToilet": true,
  "remarks": "string"
}
```

PUT /api/Hostels/{id}

Request

Headers: Content-Type: application/json

Path Parameter:

id (integer, required) - The ID of the hostel to be updated.

Request Body:

```
{
  "hostelType": "string",
  "noOfRoomsInHostel": 0,
  "noOfSeats": 0,
  "areaCoveredByHostel": 0,
  "buildingId": 0,
  "hasPlayground": true,
  "hasInternet": true,
  "hasDrinkingWater": true,
  "hasToilet": true,
  "remarks": "string"
}
```

Response (200)

Description: Hostel details updated successfully.

Response Body:

```
{
  "id": 0,
  "hostelType": "string",
  "noOfRoomsInHostel": 0,
  "noOfSeats": 0,
  "areaCoveredByHostel": 0,
  "buildingId": 0,
  "hasPlayground": true,
  "hasInternet": true,
  "hasDrinkingWater": true,
  "hasToilet": true,
  "remarks": "string"
}
```

Error Responses

400 Bad Request

- **Cause:** Malformed request or invalid data.

- **Example Response:**

```
{  
  "error": "Invalid request format."  
}
```

404 Not Found

Cause: Requested hostel does not exist.

Example Response:

```
{  
  "error": "Hostel not found."  
}
```

500 Internal Server Error

Cause: Unexpected server issue.

Example Response:

```
{  
  "error": "An unexpected error occurred. Please try again later."  
}
```

Notes

- These endpoints facilitate managing hostel records.
- Ensure that accurate details are provided when adding or updating hostel information.

5.1.4. API Documentation: Lab Management

Endpoints:

- GET /api/Labs
- POST /api/Labs
- GET /api/Labs/{id}
- PUT /api/Labs/{id}

Description:

These endpoints allow retrieving, adding, and updating lab details in the system.

GET /api/Labs

Request

- **Headers:** Authorization: `Bearer {Token},
- **Query Parameters:** None.
- **Request Body:** Not applicable.

Response (200)

Description: Returns a list of labs.

Content-Type: application/json, text/json, text/plain

Response Body:

```
[
  {
    "id": 0,
    "labName": "string",
    "buildingId": 0,
    "buildingName": "string",
    "areaCoveredByLab": 0,
    "labType": "string",
    "adequencyOfLabEquipment": true,
    "hasInternetConnection": true,
    "equipmentAtLab": "string",
    "remarks": "string"
  }
]
```

POST /api/Labs

Request

Headers: Content-Type: application/json, **Authorization: Bearer**

Request Body:

```
{
  "labName": "string",
```

```
"buildingId": 0,
"areaCoveredByLab": 0,
"labType": "string",
"adequencyOfLabEquipment": true,
"hasInternetConnection": true,
"equipmentAtLab": "string",
"remarks": "string"
}
```

Response (201)

Description: New lab added successfully.

Response Body:

```
{ "id": 0,
  "labName": "string",
  "buildingId": 0,
  "areaCoveredByLab": 0,
  "labType": "string",
  "adequencyOfLabEquipment": true,
  "hasInternetConnection": true,
  "equipmentAtLab": "string",
  "remarks": "string"
}
```

GET /api/Labs/{id}

Request

- **Headers: Authorization: Bearer**
- **Path Parameter:**
 - id (integer, required) - The ID of the lab to retrieve.

Response (200)

Description: Returns lab details.

Response Body:

```
{
  "id":0,
  "labName": "string",
  "buildingId": 0,
  "areaCoveredByLab": 0,
  "labType": "string",
  "adequencyOfLabEquipment": true,
  "hasInternetConnection": true,
  "equipmentAtLab": "string",
  "remarks": "string"
}
```

PUT /api/Labs/{id}

Request

Headers: Content-Type: application/json

Path Parameter:

id (integer, required) - The ID of the lab to be updated.

Request Body:

```
{
  "labName": "string",
  "buildingId": 0,
  "areaCoveredByLab": 0,
  "labType": "string",
  "adequencyOfLabEquipment": true,
  "hasInternetConnection": true,
  "equipmentAtLab": "string",
  "remarks": "string"
}
```

Response (200)

Description: Lab details updated successfully.

Response Body:

```
{ "id":0,
```

```
"labName": "string",
"buildingId": 0,
"areaCoveredByLab": 0,
"labType": "string",
"adequencyOfLabEquipment": true,
"hasInternetConnection": true,
"equipmentAtLab": "string",
"remarks": "string"
}
```

Error Responses

400 Bad Request

Cause: Malformed request or invalid data.

Example Response:

```
{
  "error": "Invalid request format."
}
```

404 Not Found

Cause: Requested lab does not exist.

Example Response:

```
{
  "error": "Lab not found."
}
```

500 Internal Server Error

Cause: Unexpected server issue.

Example Response:

```
{
  "error": "An unexpected error occurred. Please try again later."
}
```

Notes

- These endpoints facilitate managing lab records.

- Ensure that accurate details are provided when adding or updating lab information.

5.1.5. API Documentation: Land Management

Endpoints:

- GET /api/Lands
- POST /api/Lands
- GET /api/Lands/{id}
- PUT /api/Lands/{id}

Description:

These endpoints allow retrieving, adding, and updating land details in the system.

GET /api/Lands

Request

- **Headers: Authorization: Bearer**
- **Query Parameters:** None.
- **Request Body:** Not applicable.

Response (200)

Description: Returns a list of lands.

Content-Type: application/json, text/json, text/plain

Response Body:

```
[
  {
    "id": 0,
    "totalArea": 0,
    "unit": "string",
    "sheetNo": "string",
    "kittaNo": "string",
    "ownerShip": true,
    "remarks": "string"
  }
]
```

POST /api/Lands

Request

Headers: Content-Type: application/json, **Authorization:** Bearer

Request Body:

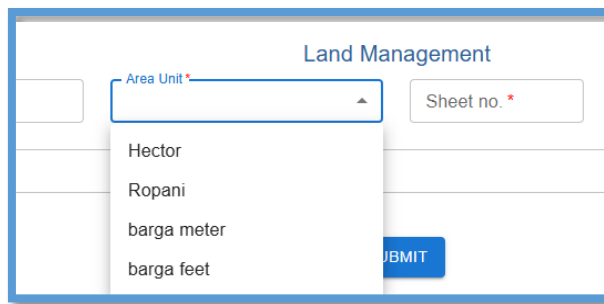
```
{  
  "id": 0,  
  "totalArea": 0,  
  "unit": "string",  
  "sheetNo": "string",  
  "kittaNo": "string",  
  "ownerShip": true,  
  "remarks": "string"  
}
```

Response (201)

Description: New land record added successfully.

Response Body:

```
{  
  "id": 0,  
  "totalArea": 0,  
  "unit": "string",  
  "sheetNo": "string",  
  "kittaNo": "string",  
  "ownerShip": true,  
  "remarks": "string"  
}
```



GET /api/Lands/{id}

Request

- **Headers: Authorization: Bearer**
- **Path Parameter:**
 - id (integer, required) - The ID of the land record to retrieve.

Response (200)

Description: Returns land record details.

Response Body:

```
{
  "id": 0,
  "totalArea": 0,
  "unit": "string",
  "sheetNo": "string",
  "kittaNo": "string",
  "ownerShip": true,
  "remarks": "string"
}
```

PUT /api/Lands/{id}

Request

Headers: Content-Type: application/json, **Authorization: Bearer**

Path Parameter:

- id (integer, required) - The ID of the land record to be updated.

Request Body:

```
{
  "id": 0,
  "totalArea": 0,
  "unit": "string",
  "sheetNo": "string",
  "kittaNo": "string",
  "ownerShip": true,
  "remarks": "string"
}
```

Response (200)

Description: Land record updated successfully.

Response Body:

```
{
  "id": 0,
  "totalArea": 0,
  "unit": "string",
  "sheetNo": "string",
  "kittaNo": "string",
  "ownerShip": true,
  "remarks": "string"
}
```

Error Responses*400 Bad Request*

Cause: Malformed request or invalid data.

Example Response:

```
{
  "error": "Invalid request format."
}
```

```
}
```

404 Not Found

Cause: Requested land record does not exist.

Example Response:

```
{  
  "error": "Land record not found."  
}
```

500 Internal Server Error

Cause: Unexpected server issue.

Example Response:

```
{  
  "error": "An unexpected error occurred. Please try again later."  
}
```

Notes

- These endpoints facilitate managing land records.
- Ensure that accurate details are provided when adding or updating land information.

5.1.6. API Documentation: Library Management

Endpoints:

- GET /api/Library
- POST /api/Library
- GET /api/Library/{id}
- PUT /api/Library/{id}

Description:

These endpoints allow retrieving, adding, and updating library details in the system.

GET /api/Library

Request

- **Headers: Authorization: Bearer**
- **Query Parameters:** None.
- **Request Body:** Not applicable.

Response (200)

Description: Returns a list of libraries.

Content-Type: application/json, text/json, text/plain

Response Body:

```
[
  {
    "id": 0,
    "libraryName": "string",
    "buildingId": 0,
    "buildingName": "string",
    "campusId": 0,
    "campusName": "string",
    "noOfLibraryRooms": 0,
    "totalAreaOfLibraryRoom": 0,
    "noOfReadingHalls": 0,
    "areaOfReadingHall": 0,
    "noOfBooks": 0,
    "noOfDailyJournals": 0,
    "noOfWeeklyJournals": 0,
    "noOfMonthlyJournals": 0,
    "noOfAnnualJournals": 0,
    "hasELibraryFacility": true,
    "remarks": "string"
  }
]
```

POST /api/Library

Request

Headers: Content-Type: application/json, **Authorization: Bearer**

Request Body:

```
{
  "libraryName": "string",
  "buildingId": 0,
  "campusId": 0,
  "noOfLibraryRooms": 0,
  "totalAreaOfLibraryRoom": 0,
  "noOfReadingHalls": 0,
  "areaOfReadingHall": 0,
  "noOfBooks": 0,
  "noOfDailyJournals": 0,
  "noOfWeeklyJournals": 0,
  "noOfMonthlyJournals": 0,
  "noOfAnnualJournals": 0,
  "hasELibraryFacility": true,
  "remarks": "string"
}
```

Response (201)

Description: New library added successfully.

Response Body:

```
{
  "id": 0,
  "libraryName": "string",
  "buildingId": 0,
  "campusId": 0,
  "noOfLibraryRooms": 0,
  "totalAreaOfLibraryRoom": 0,
  "noOfReadingHalls": 0,
  "areaOfReadingHall": 0,
  "noOfBooks": 0,
  "noOfDailyJournals": 0,
  "noOfWeeklyJournals": 0,
  "noOfMonthlyJournals": 0,
}
```

```
"noOfAnnualJournals": 0,  
"hasELibraryFacility": true,  
"remarks": "string"  
}
```

GET /api/Library/{id}

Path Parameters:

id (integer, required) -

Response (200)

- **Description:** Returns library details.

Response Body:

```
{  
  "id":0,  
  "libraryName": "string",  
  "buildingId": 0,  
  "campusId": 0,  
  "noOfLibraryRooms": 0,  
  "totalAreaOfLibraryRoom": 0,  
  "noOfReadingHalls": 0,  
  "areaOfReadingHall": 0,  
  "noOfBooks": 0,  
  "noOfDailyJournals": 0,  
  "noOfWeeklyJournals": 0,  
  "noOfMonthlyJournals": 0,  
  "noOfAnnualJournals": 0,  
  "hasELibraryFacility": true,  
  "remarks": "string"  
}
```

PUT api/Library/{id}

Path Parameters:

id (integer, required) - The ID of the employee position.

Request Body:

```
{
  "libraryName": "string",
  "buildingId": 0,
  "campusId": 0,
  "noOfLibraryRooms": 0,
  "totalAreaOfLibraryRoom": 0,
  "noOfReadingHalls": 0,
  "areaOfReadingHall": 0,
  "noOfBooks": 0,
  "noOfDailyJournals": 0,
  "noOfWeeklyJournals": 0,
  "noOfMonthlyJournals": 0,
  "noOfAnnualJournals": 0,
  "hasELibraryFacility": true,
  "remarks": "string"
}
```

Response (200)

Description: Library details updated successfully.

Response Body:

```
{
  "id":0,
  "libraryName": "string",
  "buildingId": 0,
  "campusId": 0,
  "noOfLibraryRooms": 0,
  "totalAreaOfLibraryRoom": 0,
  "noOfReadingHalls": 0,
  "areaOfReadingHall": 0,
```

```
"noOfBooks": 0,  
"noOfDailyJournals": 0,  
"noOfWeeklyJournals": 0,  
"noOfMonthlyJournals": 0,  
"noOfAnnualJournals": 0,  
"hasELibraryFacility": true,  
"remarks": "string"  
}
```

Error Responses

400 Bad Request

Cause: Malformed request or invalid data.

Example Response:

```
{  
  "error": "Invalid request format."  
}
```

404 Not Found

Cause: Requested library does not exist.

Example Response:

```
{  
  "error": "Library not found."  
}
```

500 Internal Server Error

Cause: Unexpected server issue.

Example Response:

```
{  
  "error": "An unexpected error occurred. Please try again later."  
}
```

Notes

- These endpoints facilitate managing library records.
- Ensure that accurate details are provided when adding or updating library information.

5.2. API documentation for BATCH

5.2.1. Introduction

API Documentation to get all Batch:

5.2.2. Get Batch

GET /api/Batch

Response Body:

```
[
  {
    "id": 21,
    "name": "2035",
    "batchNepali": "2090",
    "index": -4,
    "isActive": true
  },
  {
    "id": 1,
    "name": "2018",
    "batchNepali": "2075",
    "index": 1,
    "isActive": false
  },
  {
    "id": 14,
    "name": "2031",
    "batchNepali": "2088",
    "index": 14,
    "isActive": false
  }
]
```

5.3. API documentation for Fiscal Year

5.3.1. Introduction

API to get all fiscal years available in the HEMIS centralized server.

Authentication: None

GET /api/FiscalYear

Response Body:

```
[
  {
    "id": 19,
    "yearNepali": "2084/085",
    "yearEnglish": "2026/027",
    "activeFiscalYear": false,
    "index": 0
  },
  {
    "id": 18,
    "yearNepali": "2082/083",
    "yearEnglish": "2025/026",
    "activeFiscalYear": false,
    "index": 7
  }
]
```

5.4. API documentation to get Program

5.4.1. Introduction

API will get all programs assigned to a particular campus from the centralized HEMIS.

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

5.4.2. Get Program assign for campus

GET /api/ProgramMgmt/GetCollegePrograms

Response Body:

```
[
  {
    "id": 30,
    "universityId": 1,
    "campusId": null,
    "levelId": 5,
    "levelName": "Master",
    "facultyId": 8,
    "facultyName": "Management",
    "programType": "annual",
    "programId": 30,
    "programName": "Master in Business Management",
    "shortName": "MBM",
    "code": "11",
    "duration": "2",
    "alias": null,
    "remarks": null,
    "status": true
  }
]
```

5.5.API to get address for student register, employee register

1.1.1. Introduction

API will get all address as per federal structure of the country (the data inline with CBS data).

1.1.2. Get address as per federal structure

GET /api/Address/GetAll

User will get all the address as per federal structure. You are requested to use this GET API to get province, district, local level and ward no as complete address.

Response body

```
[
  {
    "id": 1,
    "combinedCode": "10101",
    "provinceCode": "1",
    "province": "Koshi",
    "provinceNepali": "कोशी प्रदेश",
    "districtCode": "01",
    "district": "Taplejung",
    "districtNepali": "ताप्लेजुङ्ग",
    "localLevelCode": "01",
    "localLevel": "Phaktanlung Rural Municipality",
    "localLevelNepali": "फक्ताङलुङ गाउँपालिका",
    "ecologicalBelt": "Mountain",
    "noOfWard": 7
  },
  {
    .
    .
    .
  },
]
```

You should send the address detail as given above.

5.6. Register/Add Department and Section for campus

This API help to create and manage department (will be used for teaching staff) and section (will be used for non-teaching staff to be categorized). User can register the department and/or section available in the campus to use for.

5.6.1. Register/Add Department for campus

Each campus may have their own department.

GET api/Management/Department

Response body

```
[
{
  "id": 10,
  "departmentName": "Department of Education",
  "status": true,
  "remarks": ""
},
{
  "id": 11,
  "departmentName": "Department of Management",
  "status": true,
  "remarks": ""
},]

```

POST api/Management/AddDepartment

Response Body:

```
{
  "departmentName": "Name of department",
  "status": true,
  "remarks": "string",
}
```

PUT api/Management/UpdateDepartment

Request

Headers: Content-Type: application/json

Path Parameter: id (integer, required)

```
{
  "id": 0,
  "departmentName": "string",
  "status": true,
  "remarks": "string"
}
```

5.6.1. Register/Add Department for campus

This API help to create/manage section for concerned campus

POST api/Managent/AddSection

Response Body:

```
{
  "SectionName": "Name of department",
  "status": true,
  "remarks": "string"
}
```

GET api/Managent/Section

Response body

```
[
  {
    "id": 6, "sectionName":
    "Administration", "status": true,
    "remarks": "Admin"
  },
  {
    "id": 7,
    "sectionName": "EMIS/IT",
    "status": true,
    "remarks": "Student emis and IT"
  },
]
```

]

PUT api/Management/UpdateSection

Request

Headers: Content-Type: application/json

Path Parameter: id (integer, required) -

```
{  
  
  "id": 0,  
  "sectionName": "string",  
  "status": true,  
  "remarks": "string"  
}
```

5.8. API Documentation for Student Management

5.8.1. Introduction

The Student API is designed to manage student-related operations within an educational system. This API supports CRUD operations for student data, handling enrollments, upgrades, and education records.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

5.8.2. Register Student

Create a New Student Record

POST /api/Student/Create

Request Body: (multipart/form-data)

```
{
  "firstName": "KALPANA",
  "middleName": "KUMARI",
  "lastName": "UPADHAYAYA",
  "phoneNumber": null,
  "gender": "female",
  "email": null,
  "admissionYearId": 25,
  "hasScholarship": false,
  "isSameAsPermanent": true,
  "rollNo": null,
  "campusId": 12,
  "nepaliName": "कल्पना कुमारी उपाध्याय",
  "doBBS": "2061/07/06", //Nepali date in yyyy/mm/dd format only
  "doBAD": "2024-01-01", //English date in yyyy-mm-dd format only
  "ethnicity": "Brahman",
  "nationality": "nepali",
  "disabilityStatus": "able",
```

```
"disabilityType": null,
"citizenshipNo": null,
"citizenIssueDist": null,
"citizenshipFront": null,
"citizenshipBack": null,
"nidno": null,
"nidPic": null,
"ppSizePhoto": null,
"pProvince": "Karnali",
"pDistrict": "Kalikot",
"pLocalLevel": "Naraharinath Rural Municipality",
"pWardNo": 3,
"pBlockNo": 0,
"pHouseNo": null,
"pLocality": null,
"tProvince": "Karnali",
"tDistrict": "Kalikot",
"tLocalLevel": "Naraharinath Rural Municipality",
"tWardNo": 3,
"tBlockNo": 0,
"tHouseNo": null,
"tLocality": null,
"fatherName": "Dashrath Bhattarai Upadhayaya",
"fatherOccupation": null,
"fatherPhoneNo": null,
"fatherEmail": null,
"motherName": null,
"motherOccupation": null,
"motherPhoneNo": null,
"motherEmail": null,
"guardianName": "Dashrath Bhattarai Upadhayaya",
"guardianOccupation": null,
"guardianPhone": "9742226166",
"guardianAddress": "Naraharinath-3,Kalikot",
"guardianEmail": null,
"programId": 37,
"programName": "Bachelor in Education",
"levelName": "Bachelor",
"facultyName": "Education",
"completionYear": 2084, //This should be Nepali year with 4 digits
```

```
"dateOfEnrollment": "2025-01-08",
"fiscalYearId": 7,
"edg": false,
"semester": null,
"year": "First",
"programType": "annual",
"section": null,
"type": null,
"isMuktaKamaiya": false,
"isFromMartyrFamily": false,
"entranceScore": null,
"entranceSymbol": null,
"digitalSignature": null,
"regNoManual": null,
"religion": "Hinduism",
"rollNoManual": null,
"universityRegdNo": null,
"medium": null,
"shift": null
}
```

Response Body

```
{
  "id": 3598,
  "firstName": "KALPANA",
  "middleName": "KUMARI",
  "lastName": "UPADHAYAYA",
  "phoneNumber": null,
  "gender": "female",
  "email": null,
  "admissionYearId": 25,
  "hasScholarship": false,
  "isSameAsPermanent": true,
  "rollNo": null,
  "collegeName": "Simta Multiple Campus",
```

```
"uniName": "",
"campusId": 12,
"nepaliName": "कल्पना कुमारी उपाध्याय",
"doBBS": "2061/07/27",
"doBAD": "2024-01-01T00:00:00",
"ethnicity": "Brahman",
"nationality": "nepali",
"disabilityStatus": "able",
"disabilityType": null,
"citizenshipNo": null,
"citizenIssueDist": null,
"citizenshipFront": null,
"citizenshipBack": null,
"nidno": null,
"nidPic": null,
"ppSizePhoto": null,
"semesterOrYearNo": null,
"pProvince": "Karnali",
"pDistrict": "Kalikot",
"pLocalLevel": "Naraharinath Rural Municipality",
"pWardNo": 3,
"pBlockNo": 0,
"pHouseNo": null,
"pLocality": null,
"tProvince": "Karnali",
"tDistrict": "Kalikot",
"tLocalLevel": "Naraharinath Rural Municipality",
"tWardNo": 3,
"tBlockNo": 0,
"tHouseNo": null,
"tLocality": null,
```

"fatherName": "Dashrath Bhattarai Upadhayaya",
"fOccupation": null,
"fatherPhoneNo": null,
"fatherEmail": null,
"motherName": null,
"mOccupation": null,
"motherPhoneNo": null,
"motherEmail": null,
"guardianName": "Dashrath Bhattarai Upadhayaya",
"guardianOccupation": null,
"guardianPhone": "9742226166",
"gAddress": "Naraharinath-3,Kalikot",
"gEmail": null,
"programId": 37,
"programName": "Bachelor in Education",
"levelName": "Bachelor",
"facultyName": "Education",
"fiscalYear": "2081/082",
"admissionYear": "0",
"completionYear": 2028,
"dateOfEnrollment": "2025-01-08",
"fiscalYearId": 7,
"institutionId": null,
"isVerified": true,
"programfee": null,
"paidAmount": null,
"scholarshipAmt": null,
"receiptImage": null,
"edg": false,
"semester": null,
"year": "First",

```
"programType": "annual",
"section": null,
"batchNameNepali": "2081",
"batchName": null,
"batchId": 0,
"userId": 0,
"universityId": 0,
"type": null,
"isMuktaKamaiya": false,
"isFromMartyrFamily": false,
"entranceScore": null,
"entranceSymbol": null,
"digitalSignature": null,
"regNoManual": null,
"religion": "Hinduism",
"rollNoManual": null,
"universityRegdNo": null,
"isRegNoManual": true,
"isRollNoManual": true,
"medium": null,
"shift": null
},
```

Response:

```
{
  "status": "Success",
  "message": "Student Registered Successfully."
}
```

Error Handling

The API returns standard HTTP response codes to indicate success or failure.

Common Status Codes:

- 200 OK - The request was successful.
- 400 Bad Request - The request contained invalid parameters.
- 401 Unauthorized - Authentication failed or missing token.
- 404 Not Found - The requested resource was not found.
- 500 Internal Server Error - A server error occurred.

Endpoints

5.8.3. Get All Students

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

Retrieve a List of All Students

GET /api/Student/GetAllStudent

Query Parameters:

- isVerified (boolean, optional) - Filter students by verification status.
- UniId (integer, optional) - Filter students by university ID.

Response:

```
{
  "status": "Success",
  "data":
  [
    {
      "id": 82,
      "firstName": "HARI",
      "lastName": "BASNET",
      "phoneNumber": "9810909899",
```

```
"gender": "male",
"email": "studentemail@gmail.com",
"admissionYearId": 6,
"rollNo": "1-10-2018-0004",
"collegeName": "Your Campus Name",
"nepaliName": "हरी बस्नेत",
"doBBS": "2003/05/03",
"ethnicity": "Hindu",
"nationality": "Nepali",
"pProvince": "Koshi",
"pDistrict": "Okhaldhunga",
"tProvince": "Koshi",
"tDistrict": "Okhaldhunga",
"guardianName": "Xander Griffin",
"guardianPhone": "8987879008",
"programId": 3,
"programName": "Master in Mass Communication and Journalism",
"levelName": "Master",
"facultyName": "Humanities and Social Sciences",
"fiscalYear": "2081/082",
"admissionYear": "0",
"completionYear": 2082,
"dateOfEnrollment": "2025-03-02",
"isVerified": true,
"semester": "First",
"programType": "semester",
"batchNameNepali": "2080",
"isMuktaKamaiya": true,
"isFromMartyrFamily": false
}
```

]

5.8.4. Get Student by ID

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

Retrieve details of a student by ID

GET /api/Student/{studentId}

Path Parameters:

- studentId (integer, required) - The ID of the student.

Response:

```
{
  "id": 82,
  "firstName": "HARI",
  "lastName": "BASNET",
  "phoneNumber": "9810909899",
  "gender": "male",
  "email": "studentemail@gmail.com",
  "admissionYearId": 6,
  "rollNo": "1-10-2018-0004",
  "collegeName": "Your Campus Name",
  "nepaliName": "हरी बस्नेत",
  "doBBS": "2003/05/03",
  "ethnicity": "Hindu",
  "nationality": "Nepali",
  "pProvince": "Koshi",
  "pDistrict": "Okhaldhunga",
  "tProvince": "Koshi",
  "tDistrict": "Okhaldhunga",
  "guardianName": "Xander Griffin",
```

```
"guardianPhone": "9810909899",
"programId": 3,
"programName": "Master in Mass Communication and Journalism",
"levelName": "Master",
"facultyName": "Humanities and Social Sciences",
"fiscalYear": "2081/082",
"admissionYear": "0",
"completionYear": 2082,
"dateOfEnrollment": "2025-03-02",
"isVerified": true,
"semester": "First",
"programType": "semester",
"batchNameNepali": "2080",
"isMuktaKamaiya": true,
"isFromMartyrFamily": false
}
```

5.8.5. Update Student

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

Update an existing student record

PATCH /api/Student/{studentId}

Path Parameters:

- studentId (integer, required) - The ID of the student to update.

Request Body: (multipart/form-data)

```
{
  "id": 3598,
  "firstName": "KALPANA",
}
```

```
"middleName": "KUMARI",
"lastName": "UPADHAYAYA",
"phoneNumber": null,
"gender": "female",
"email": null,
"admissionYearId": 25,
"hasScholarship": false,
"isSameAsPermanent": true,
"rollNo": null,
"campusId": 12,
"nepaliName": "कल्पना कुमारी उपाध्याय",
"doBBS": "2061/07/06", //Nepali date in yyyy/mm/dd format only
"doBAD": "2024-01-01", //English date in yyyy-mm-dd format only
"ethnicity": "Brahman",
"nationality": "nepali",
"disabilityStatus": "able",
"disabilityType": null,
"citizenshipNo": null,
"citizenIssueDist": null,
"citizenshipFront": null,
"citizenshipBack": null,
"nidno": null,
"nidPic": null,
"ppSizePhoto": null,
"pProvince": "Karnali",
"pDistrict": "Kalikot",
"pLocalLevel": "Naraharinath Rural Municipality",
"pWardNo": 3,
"pBlockNo": 0,
"pHouseNo": null,
"pLocality": null,
```

"tProvince": "Karnali",
"tDistrict": "Kalikot",
"tLocalLevel": "Naraharinath Rural Municipality",
"tWardNo": 3,
"tBlockNo": 0,
"tHouseNo": null,
"tLocality": null,
"fatherName": "Dashrath Bhattarai Upadhayaya",
"fOccupation": null,
"fatherPhoneNo": null,
"fatherEmail": null,
"motherName": null,
"mOccupation": null,
"motherPhoneNo": null,
"motherEmail": null,
"guardianName": "Dashrath Bhattarai Upadhayaya",
"guardianOccupation": null,
"guardianPhone": "9742226166",
"gAddress": "Naraharinath-3,Kalikot",
"gEmail": null,
"programId": 37,
"programName": "Bachelor in Education",
"levelName": "Bachelor",
"facultyName": "Education",
"completionYear": 2084, //This should be Nepali year with 4 digits
"dateOfEnrollment": "2025-01-08",
"fiscalYearId": 7,
"edg": false,
"semester": null,
"year": "First",
"programType": "annual",

```
"section": null,
"type": null,
"isMuktaKamaiya": false,
"isFromMartyrFamily": false,
"entranceScore": null,
"entranceSymbol": null,
"digitalSignature": null,
"regNoManual": null,
"religion": "Hinduism",
"rollNoManual": null,
"universityRegdNo": null,
"medium": null,
"shift": null
}
```

Response:

```
{
  "status": "Success",
  "message": "Student record updated successfully."
}
```

Error Handling

The API returns standard HTTP response codes to indicate success or failure.

Common Status Codes:

- 200 OK - The request was successful.
- 400 Bad Request - The request contained invalid parameters.
- 401 Unauthorized - Authentication failed or missing token.
- 404 Not Found - The requested resource was not found.
- 500 Internal Server Error - A server error occurred.

5.9.API for student Upgrade

5.9.1. Introduction

Student Upgrade: This API is used to upgrade students from the current semester/year to another semester/year.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

5.9.2. Upgrade Students

Upgrade students from one semester/year to next

POST /api/StudentUpgrade

POST Parameters:

- studentId (integer, required) - The ID of the student to update.

Request Body: (application/json)

```
{
  "Id":0,
  "programId": 0,
  "studentId": 1,
  "year": "First", // First, Second, Third, Fourth, null
  "semester": null, // First, Second, Third, Fourth, Fifth, Sixth, Seventh, Eighth,
  null
  "batchId": 0,
  "isIrregular": false
},
{
  "Id":0,
  "programId": 0,
  "studentId": 2,
  "year": "First", // First, Second, Third, Fourth, null
  "semester": null, // First, Second, Third, Fourth, Fifth, Sixth, Seventh, Eighth,
  null
  "batchId": 0,
  "isIrregular": false
}
```

Response:

```
{
```

```
"id": 4269,
"programId": 37,
"programName": "Bachelor in Education",
"programType": "annual",
"levelName": "Bachelor",
"facultyName": "Education",
"studentId": 4126,
"student": {
  "id": 4126,
  "userId": null,
  "programId": 37,
  "firstName": "SANTOSH",
  "middleName": null,
  "lastName": "B. K.",
  "phoneNumber": "9742824223",
  "gender": "male",
  "email": null,
  "rollNo": null,
  "universityId": null,
  "campusId": 42,
  "nepaliName": "सन्तोष बि. क.",
  "doBBS": "2061/10/02",
  "doBAD": "0001-01-01T00:00:00",
  "ethnicity": "Dalit",
  "nationality": "nepali",
  "disabilityStatus": "able",
  "disabilityType": null,
  "citizenshipNo": null,
  "citizenIssueDist": null,
  "citizenshipFront": null,
  "citizenshipBack": null,
  "nidno": null,
  "nidPic": null,
  "ppSizePhoto": null,
  "pProvince": "Karnali",
  "pDistrict": "Surkhet",
  "pLocalLevel": "Simta Rural Municipality",
  "pWardNo": 4,
  "pBlockNo": 0,
  "pHouseNo": null,
  "pLocality": null,
  "tProvince": "Karnali",
  "tDistrict": "Surkhet",
  "tLocalLevel": "Simta Rural Municipality",
  "tWardNo": 4,
  "tBlockNo": 0,
```

```
"tHouseNo": null,
"tLocality": null,
"fatherName": null,
"fOccupation": null,
"fatherPhoneNo": null,
"fatherEmail": null,
"motherName": null,
"mOccupation": null,
"motherPhoneNo": null,
"motherEmail": null,
"guardianName": "Amrit Bahadur Nepali",
"guardianOccupation": null,
"guardianPhone": "9742824223",
"guardianAddress": "Simta-4 Surkhet",
"guardianEmail": null,
"type": null,
"semesterOrYearNo": null,
"admissionYear": 0,
"completionYear": 2027,
"dateOfEnrollment": "2023-12-27",
"fiscalYearId": 6,
"institutionId": null,
"isVerified": true,
"programfee": null,
"paidAmount": null,
"scholarshipAmt": null,
"receiptImage": null,
"edg": false,
"semester": null,
"year": null,
"programType": null,
"section": null,
"batchNameNepali": null,
"batchName": null,
"batchId": 0,
"fiscalYearName": "2080/081",
"religion": "Hinduism",
"universityRegdNo": null,
"isRegNoManual": true,
"isRollNoManual": true,
"medium": null,
"shift": null,
"rollNoManual": null
},
"year": "First",
"semester": "",
```

```
"batch": "2023",  
"batchNepali": "2080",  
"batchId": 24,  
"fiscalYear": "2080/081"  
}  
]
```

5.10. API Document for Employee Position

5.10.1. Introduction

Users need to use the following API to register and manage employee data. To register employee user should get the employee position for which the below API is used.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

Endpoints

Employee Positions

Retrieve a List of Employee Positions

GET /api/EmployeePositions

Response:

```
[
  {
    "id": 1,
    "category": "Nonteaching",
    "type": "Administrative",
    "postName": "Ass. Accountant",
    "index": 1,
    "status": true,
    "remarks": ""
  }
]
```

Get Employee Position by ID

GET /api/EmployeePositions/{id}

Path Parameters:

- id (integer, required) - The ID of the employee position.

Response:

```
{
  "id": 1,
  "category": "Nonteaching",
  "type": "Administrative",
  "postName": "Ass. Accountant",
  "index": 1,
  "status": true,
  "remarks": ""
}
```

Error Handling

The API returns standard HTTP response codes to indicate success or failure.

Common Status Codes:

- 200 OK - The request was successful.
- 400 Bad Request - The request contained invalid parameters.
- 401 Unauthorized - Authentication failed or missing token.
- 404 Not Found - The requested resource was not found.
- 500 Internal Server Error - A server error occurred.

5.11. **API Documentation: Employee Management**

5.11.1. **Introduction**

Users need to use the following API to register and manage employee data.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

5.11.2. **Get Employee**

Retrieve a List of Employee Positions

GET /api/Employee/GetAllEmployeees

Response:

```
[
{
  "id": 240,
  "userId": null,
  "user": null,
  "firstName": "SAN",
  "middleName": "BAHADUR",
  "lastName": "BUDHA",
  "phoneNumber": "9748314286",
  "gender": "male",
  "email": "budha2023@gmail.com",
  "ethnicity": "Chhetri",
  "campusId": 12,
  "universityId": null,
  "teachingFacultyName": "undefined",
  "employeeType": "Other",
  "citizenshipNo": "650",
  "orgId": null,
  "dateOFBirth": "2023/04/09",
  "dateOFBirthAd": "2023-04-06",
  "employeePositionId": 55,
  "postName": "Peon",
  "positionType": "Other",
  "salutation": "Mr.",
  "ctzIssueDistrictCode": null,
  "nidNo": null,
  "citizenshipFront": null,
  "citizenshipBack": null,
  "nidPage": null,
  "ppSizePhoto": null,
```

"pProvince": "Karnali",
"pDistrict": "Surkhet",
"pLocalLevel": "Simta Rural Municipality",
"pWardNo": 6,
"pTole": null,
"pHouseNo": null,
"pAddress": null,
"fiscalYearId": null,
"tProvince": "Karnali",
"tDistrict": "Surkhet",
"tLocalLevel": "Simta Rural Municipality",
"tWardNo": 6,
"tHouseNo": null,
"tTole": null,
"tAddress": null,
"group": "undefined",
"subGroup": null,
"position": "Nonteaching",
"joiningType": "permanent",
"joiningDate": "2070/08/24",
"workPhone": null,
"workEmail": null,
"referenceLetter": null,
"joiningLetter": null,
"otherLetter": null,
"nonTeachingType": null,
"institutionId": 42,
"graduatedFrom": "Bhanu Secondary School",
"facultyName": "undefined",
"levelName": null,
"programName": "TSLC",

```
"enrolledYear": null,
"passedYear": null,
"certificateCopy": null,
"transCriptCopy": null,
"marksheetCopy": null,
"otherImage": null,
"collegeName": "Simta Multiple Campus",
"campusName": null,
"bankAc": "08020022538",
"bankName": "Laxmi Sunrise Bank ",
"bankBranch": "Simta Surkhet",
"panNo": "138880088",
"departmentId": null,
"sectionId": 50,
"isSameAsPermanent": false,
"religion": "Hinduism"
}
]
```

5.11.3. POST Employee

POST /api/Employee

Request Body:

```
{
  "firstName": "SAN",
  "middleName": "BAHADUR",
  "lastName": "BUDHA",
  "phoneNumber": "9748314286",
  "gender": "male", // male, female
  "email": "budha2023@gmail.com",
  "ethnicity": "Chhetri",
```

```
"campusId": 12,
"teachingFacultyName": "undefined",
"employeeType": "Other", // Type is = ( teaching, Administrative, Technical,
                          Other)
"citizenshipNo": "650",
"dateOFBirth": "2023/04/07",
"dateOFBirthAd": "2000-05-06",
"employeePositionId": 55, // See GET /api/EmployeePositions
"postName": "Peon",
"positionType": "Other",
"salutation": "Mr.", // value= (Dr., Miss, Mr., Mrs., Professor, )
"ctzIssueDistrictCode": null,
"nidNo": null,
"citizenshipFront": null,
"citizenshipBack": null,
"nidPage": null,
"ppSizePhoto": null,
"pProvince": "Karnali",
"pDistrict": "Surkhet",
"pLocalLevel": "Simta Rural Municipality",
"pWardNo": 6,
"pTole": null,
"pHouseNo": null,
"pAddress": null,
"fiscalYearId": null,
"tProvince": "Karnali",
"tDistrict": "Surkhet",
"tLocalLevel": "Simta Rural Municipality",
"tWardNo": 6,
"tHouseNo": null,
"tTole": null,
```

```
"tAddress": null,
"position": "Nonteaching",
"joiningType": "permanent",
"joiningDate": "2070/08/24",
"workPhone": null,
"workEmail": null,
"referenceLetter": null,
"joiningLetter": null,
"otherLetter": null,
"nonTeachingType": null,
"institutionId": 12,
"graduatedFrom": "Bhanu Secondary School",
"facultyName": "undefined", //You should input the graduated faculty name of
                             employee
"levelName": null,
"programName": "TSLC", //Program name employee graduated from
"enrolledYear": null,
"passedYear": null,
"certificateCopy": null,
"transCriptCopy": null,
"marksheetCopy": null,
"otherImage": null,
"collegeName": "Simta Multiple Campus",
"bankAc": "08020022538",
"bankName": "Laxmi Sunrise Bank ",
"bankBranch": "Simta Surkhet",
"panNo": "138880088",
"departmentId": null, //User can ignore this field. However it's Department ID
in                          which teaching employee working for.
"sectionId": 50, //User can ignore this field. However it's section ID in which
                             non-teaching employee working for.
```

```
        "isSameAsPermanent": false,
        "religion": "Hinduism" // value =(Buddhist, Hinduism, Islamic, Kirat, Others,
                                Christian)
    }
]
```

Response:

```
{
  "status": "Success",
  "message": "Employee position created successfully."
}
```

Error Handling

The API returns standard HTTP response codes to indicate success or failure.

Common Status Codes:

- 200 OK - The request was successful.
- 400 Bad Request - The request contained invalid parameters.
- 401 Unauthorized - Authentication failed or missing token.
- 404 Not Found - The requested resource was not found.
- 500 Internal Server Error - A server error occurred.

5.11.4. Update Employee

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

Update an existing employee record

PATCH /api/Employee/{employeeId}

Path Parameters:employeeId(integer, required) - The ID of the student to update.

Request Body: (multipart/form-data)

```
{
  "id": 240,
  "firstName": "SAN",
  "middleName": "BAHADUR",
  "lastName": "BUDHA",
  "phoneNumber": "9748314286",
  "gender": "male", // male, female
  "email": "budha2023@gmail.com",
  "ethnicity": "Chhetri",
  "campusId": 12,
  "teachingFacultyName": "undefined",
  "employeeType": "Other", //Type is = ( teaching, Adminstrative, Technical,
                                Other)
  "citizenshipNo": "650",
  "dateOFBirth": "2023/04/07",
  "dateOFBirthAd": "2000-05-06",
  "employeePositionId": 55, // See GET /api/EmployeePositions THIS API return
                                the position created for employee based on the
                                employeeType ( teaching, Adminstrative, Technical,
                                Other)
  "postName": "Peon",
  "positionType": "Other",
  "salutation": "Mr.", // value= (Dr., Miss, Mr., Mrs., Professor, )
  "ctzIssueDistrictCode": null,
  "nidNo": null,
  "citizenshipFront": null,
  "citizenshipBack": null,
  "nidPage": null,
  "ppSizePhoto": null,
  "pProvince": "Karnali",
  "pDistrict": "Surkhet",
```

```
"pLocalLevel": "Simta Rural Municipality",
"pWardNo": 6,
"pTole": null,
"pHouseNo": null,
"pAddress": null,
"fiscalYearId": null,
"tProvince": "Karnali",
"tDistrict": "Surkhet",
"tLocalLevel": "Simta Rural Municipality",
"tWardNo": 6,
"tHouseNo": null,
"tTole": null,
"tAddress": null,
"position": "Nonteaching",
"joiningType": "permanent",
"joiningDate": "2070/08/24",
"workPhone": null,
"workEmail": null,
"referenceLetter": null,
"joiningLetter": null,
"otherLetter": null,
"nonTeachingType": null,
"institutionId": 12,
"graduatedFrom": "Bhanu Secondary School",
"facultyName": "undefined", //You should input the graduated faculty name of
                             employee
"levelName": null,
"programName": "TSLC", //Program name employee graduated from
"enrolledYear": null,
"passedYear": null,
"certificateCopy": null,
```

```

    "transCriptCopy": null,
    "marksheetCopy": null,
    "otherImage": null,
    "bankAc": "08020022538",
    "bankName": "Laxmi Sunrise Bank ",
    "bankBranch": "Simta Surkhet",
    "panNo": "138880088",
    "departmentId": null, //User can ignore this field. However it's Department ID
in           which teaching employee working for.
    "sectionId": 50, //User can ignore this field. However it's section ID in which
           non-teaching employee working for.
    "isSameAsPermanent": false,
    "religion": "Hinduism" // value =(Buddhist, Hinduism, Islamic, Kirat, Others,
           Christian,)
    }
]

```

Response:

```

{
  "status": "Success",
  "message": "Employee updated successfully."
}

```

Error Handling

The API returns standard HTTP response codes to indicate success or failure.

Common Status Codes:

- 200 OK - The request was successful.
- 400 Bad Request - The request contained invalid parameters.
- 401 Unauthorized - Authentication failed or missing token.
- 404 Not Found - The requested resource was not found.
- 500 Internal Server Error - A server error occurred.

5.12. API for Graduate Student

5.12.1. Introduction

Student Graduation: This API manages the data of graduated students.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

To manage data of graduated students enrolled or listed in the system.

POST /api/Graduation/GenerateGraduationApplication

'Content-Type: multipart/form-data'

Request Body:

```
{
  "ProgramMgmtId": "",
  "ContactNo": "",
  "Gender": "",
  "EnrolledYear": "",
  "DoBEng": "",
  "SymbolNoUniversity": "",
  "FatherName": "",
  "UploadOtherDoc": "",
  "ApplicantNameNep": "",
  "UniversityIssueNo": "",
  "Division": "",
  "PassedYear": "",
  "MotherName": "",
  "UploadPPSizePhoto": "",
  "ApplicantNameEng": "",
  "CampusRolNo": "",
```

```
"Remarks": "",
"StudentID": "",
"FiscalYearID": "",
"UploadTranscript": "",
"UploadReceipt": "",
"UploadSignature": "",
"GPA": "",
"StudentRegNo": "",
"CampusId": "",
"ID": "",
"DoBNepali": "",
"Email": ""
}
```

This API to post Graduation Information for the student not in the student enrolled list, meaning that the student studied earlier than the system being implemented.

POST /api/Graduation/GenerateGraduationApplicationForStudent

'Content-Type: multipart/form-data'

```
{
  "ProgramMgmtId": "",
  "ContactNo": "",
  "Gender": "",
  "EnrolledYear": "",
  "DoBEng": "",
  "SymbolNoUniversity": "",
  "FatherName": "",
  "UploadOtherDoc": "",
  "ApplicantNameNep": "",
  "UniversityIssueNo": "",
  "Division": "",
```

```
"PassedYear": "",
"MotherName": "",
"UploadPPSizePhoto": "",
"ApplicantNameEng": "",
"CampusRolNo": "",
"Remarks": "",
"StudentID": "",
"FiscalYearID": "",
"UploadTranscript": "",
"UploadReceipt": "",
"UploadSignature": "",
"GPA": "",
"StudentRegNo": "",
"CampusId": "",
"ID": "",
"DoBNepali": "",
"Email": ""
}
```

GET /api/Graduation/id

```
{
  "id": 0,
  "fiscalYearID": 0,
  "studentID": 0,
  "universityName": "string",
  "campusId": 0,
  "campusName": "string",
  "facultyName": "string",
  "levelName": "string",
}
```

```
"programType": "string",
"programName": "string",
"applicantNameEng": "string",
"applicantNameNep": "string",
"doBNepali": "2075/04/11",
"doBEng": "2025-04-11",
"email": "string",
"universityIssueNo": "string",
"studentRegNo": "string",
"symbolNoUniversity": "string",
"campusRolNo": "string",
"enrolledYear": 0,
"passedYear": 0,
"division": "string",
"gpa": 0,
"fatherName": "string",
"motherName": "string",
"contactNo": "string",
"uploadSignature": "string",
"uploadPPSizePhoto": "string",
"uploadReceipt": "string",
"uploadTranscript": "string",
"uploadOtherDoc": "string",
"remarks": "string",
"programMgmtId": 0,
"gender": "string"
}
```

PUT /api/Graduation/id

```
{
```

```
"id": 0,  
"fiscalYearID": 0,  
"studentID": 0,  
"universityName": "string",  
"campusId": 0,  
"campusName": "string",  
"facultyName": "string",  
"levelName": "string",  
"programType": "string",  
"programName": "string",  
"applicantNameEng": "string",  
"applicantNameNep": "string",  
"doBNepali": "2075/04/11",  
"doBEng": "2025-04-11",  
"email": "string",  
"universityIssueNo": "string",  
"studentRegNo": "string",  
"symbolNoUniversity": "string",  
"campusRolNo": "string",  
"enrolledYear": 0,  
"passedYear": 0,  
"division": "string",  
"gpa": 0,  
"fatherName": "string",  
"motherName": "string",  
"contactNo": "string",  
"uploadSignature": "string",  
"uploadPPSizePhoto": "string",  
"uploadReceipt": "string",  
"uploadTranscript": "string",  
"uploadOtherDoc": "string",
```

```
"remarks": "string",
"programMgmtId": 0,
"gender": "string"
}
```

5.13. API for Dropout Student

5.13.1. Introduction

Student Dropout: This API manages the data of dropout students.

Authentication

This API uses token-based authentication. All requests must include an Authorization header:

Authorization: Bearer <token>

'Content-Type: multipart/form-data'

GET /api/DropOut

Request Body:

```
[
  {
    "id": 0,
    "studentName": "string",
    "studentNameNepali": "string",
    "email": "string",
    "batch": 0,
    "rollNo": "string",
    "studentId": 0,
    "reason": "string",
    "dateOfEntry": "string",
    "remarks": "string"
  }
]
```

content-type:application/json

GET /api/DropOut/id

```
{  
  "id": 0,  
  "studentName": "string",  
  "studentNameNepali": "string",  
  "email": "string",  
  "batch": 0,  
  "rollNo": "string",  
  "studentId": 0,  
  "reason": "string",  
  "dateOfEntry": "string",  
  "remarks": "string"  
}
```

POST /api/DropOut

```
{  
  "studentId": 0,  
  "reason": "string",  
  "dateOfEntry": "string",  
  "remarks": "string"  
}
```

PUT /api/DropOut/id

```
{  
  "studentId": 0,  
  "reason": "string",  
  "dateOfEntry": "string",  
  "remarks": "string"  
}
```

}

Note: You are not permitted to implement these APIs without adhering to the guidelines prepared and published by the UGC for the preparation, establishment and implementation of HEMIS. It is the responsibility of the implementing agencies to ensure compliance with the ICT Act, relevant guidelines, strategies, and any other official documents published by the concerned ministries or departments of the Government of Nepal.

THE END