

# University Time Table Management System

A full Backend implementation of an university time table management system consistent of various number of models, controllers and routes. This development was done for the Assignment 1 of the Application Frameworks module of the 3rd Year 2nd Semester Software Engineering specialization.

**Developed By: Jayakody T.N.A. (IT21345296)**

Includes API's for:

- User Management
- Room Management
- Student Management
- Course Management
- Resource Management
- Time Table Management
- Booking Management
- Enrollment Management

---

## User & Auth API Collection

This folder includes all the API regarding the user and authentication management of the university management system. One can register (sign up) users, login users, view all users, update users, delete users, view specific users and search specific needed users as per need using the API functions in this collection.

---

### POST User Signup

`http://localhost:8070/auth/add`

Users (admins, faculty) can register themselves within the system using their information and credentials using this API function.

## Body raw (json)

---

json

```
{
  "fullName": "Computing - DS",
  "email": "compds@sliit.lk",
  "address": "DS Dept, SLIIT",
  "telephoneNumber": 771233452,
  "userType": "faculty",
  "userName": "dsdept",
  "password": "ds123"
}
```

---

## POST User Login

http://localhost:8070/auth/login

This API function allows the users (admin, faculty, student) to login to the system using their credentials and the user types.

## Body raw (json)

---

json

```
{
  "email": "heshan@gmail.com",
  "password": "hs123",
  "userType": "student"
}
```

---

## GET Get All Users

http://localhost:8070/user/

Using this API users can view all the courses available within the system with available authentication.

## AUTHORIZATION Bearer Token

---

Token <token>

---

## PUT Update User By Id

<http://localhost:8070/user/update/65ffa4ad66a7ef550d18f01f>

Users can update their user accounts via this API by using the course Id with active authentication token.

## AUTHORIZATION Bearer Token

---

Token <token>

## Body raw (json)

---

json

```
{
  "fullName": "Computing - DS",
  "email": "compds@slit.lk",
  "address": "DS Department, SLIIIT",
  "telephoneNumber": 771233452,
  "userType": "faculty",
  "userName": "dsdept",
  "password": "ds123"
}
```

## DELETE Delete User By Id

<http://localhost:8070/user/delete/65ffa4ad66a7ef550d18f01f>

Users can delete their user accounts via this API by using the course Id with active authentication token.

---

## GET View User By Id

`http://localhost:8070/user/get/65ffa49666a7ef550d18f01c`

Users can view their user accounts via this API by using the course Id with active authentication token.

## AUTHORIZATION Bearer Token

---

Token	<token>
-------	---------

---

## GET Search User By Keywords

`http://localhost:8070/user/search/admin`

Users can search other user accounts via this API by using the course Id with active authentication token.

## AUTHORIZATION Bearer Token

---

Token	<token>
-------	---------

---

## Course API Collection

This folder includes all the API regarding the **course management** of the university. One can add new courses, view all courses, update courses, delete courses, view specific courses and search specific needed courses as per need using the API functions in this collection.

---

## POST Add Course

## POST Add Course

http://localhost:8070/Course/add

This API allows users to create courses for the university by filling out the respective details.

### Body raw (json)

---

json

```
{
  "courseName": "ITP",
  "courseCode": "IT2080",
  "description": "Module with an information technology project for all Studen
  "noOfCredits": 4
}
```

---

## GET Get All Courses

http://localhost:8070/Course/

Using this API users can view all the courses available within the system.

---

## PUT Update Course By Id

http://localhost:8070/Course/update/65fe67cdeb2a4764d9bccb43

Users can update a specific course via this API by using the course Id.

### Body raw (json)

---

---

json

```
{  
  "courseName": "Information Technology Project",  
  "courseCode": "IT2080",  
  "description": "Module with an information technology project for all Students",  
  "noOfCredits": 4  
}
```

---

## DELETE Delete Course By Id

<http://localhost:8070/Course/delete/65fe6c26eb2a4764d9bccb49>

Users can delete a specific course via this API by using the course Id.

---

## GET View Course By Id

<http://localhost:8070/Course/get/65fe6790eb2a4764d9bccb41>

Users can view a specific course via this API by using the course Id.

---

## GET Search Course By Keywords

<http://localhost:8070/Course/search/Distributed>

Users can view specific courses via this API by using specific keywords regarding the course.

---

## Room API Collection

This folder includes all the API regarding the **room management** of the university. One can add new rooms, view all rooms, update rooms, delete rooms, view specific rooms and search specific needed rooms as per need using the API functions in this collection.

---

## POST Add Room

http://localhost:8070/Room/add

This API allows users to create rooms for the university by filling out the respective details.

### Body raw (json)

---

json

```
{
  "roomName": "Engineering Auditorium",
  "roomNumber": "EA01",
  "location": "Engineering Building",
  "noOfSeats": 200,
  "availability": true
}
```

---

## GET Get All Rooms

http://localhost:8070/Room/

Using this API users can view all the rooms available within the system.

---

## PUT Update Room By Id

http://localhost:8070/Room/update/65fe8ce7173b16ba4ad52b85

Users can update a specific room via this API by using the room Id.

### Body raw (json)

---

json

```
{  
  "roomName": "Engineering Auditorium",  
  "roomNumber": "EA01",  
  "location": "Engineering Building",  
  "noOfSeats": 200,  
  "availability": false  
}
```

---

## DELETE Delete Room By Id

<http://localhost:8070/Room/delete/65fe86807d9d5c54705c7f7f>

Users can delete a specific room via this API by using the room Id.

---

## GET View Room By Id

<http://localhost:8070/Room/get/65fe86a558537d754c432eea>

Users can view a specific room via this API by using the room Id.

---

## GET Search Room By Keywords

<http://localhost:8070/Room/search/G606>

Users can view specific rooms via this API by using specific keywords regarding the room.



This folder includes all the API regarding the resource management of the university. One can add new resources, view all resources, update resources, delete resources, view specific resources and search specific needed resources as per need using the API functions in this collection.

---

## POST Add Resource

<http://localhost:8070/Resource/add>

This API allows users to create resources for the university by filling out the respective details.

### Body raw (json)

---

json

```
{
  "resourceName": "Smart Board",
  "resourceCode": "SB99",
  "description": "Smart Board used to screen and edit from the board itself",
  "faculty": "Computing",
  "availability": true
}
```

---

## GET Get All Resources

<http://localhost:8070/Resource/>

Using this API users can view all the resources available within the system.

---

## PUT Update Resource By Id

<http://localhost:8070/Resource/update/CE5f2d00170b10b54e4d50b0d>

http://localhost:8070/Resource/update/65fe8d68173b16ba4ad52b8d

Users can update a specific resource via this API by using the resource Id.

### Body raw (json)

---

json

```
{
  "resourceName": "Smart Board",
  "resourceCode": "SB109",
  "description": "Smart Board used to screen and edit from the board itself",
  "faculty": "Computing",
  "availability": true
}
```

---

## DELETE Delete Resource By Id

http://localhost:8070/Resource/delete/65fe8d68173b16ba4ad52b8d

Users can delete a specific resource via this API by using the resource Id.

---

## GET View Resource By Id

http://localhost:8070/Resource/get/65fe8d58173b16ba4ad52b8b

Users can view a specific resource via this API by using the resource Id.

---

## GET Search Resource By Keywords

http://localhost:8070/Resource/search/Projector

Users can view specific resources via this API by using specific keywords regarding the resource.

---

## Booking API Collection

This folder includes all the API regarding the booking resources/rooms management of the university. One can add new bookings, view all bookings, update bookings, delete bookings, view specific bookings and search specific needed bookings as per need using the API functions in this collection.

---

### POST Add Booking

<http://localhost:8070/Booking/add>

This API allows users to create bookings by filling out the respective details.

#### Body raw (json)

---

json

```
{
  "date": "2024.04.21",
  "timeDuration": "8:30 - 11:30",
  "description": "Projector for ITP",
  "faculty": "Computing",
  "batchName": "Y2.S2.IT.WE.02.02",
  "bookingEntity": "PR567"
}
```

---

### GET Get All Bookings

<http://localhost:8070/Booking/>

Using this API users can view all the bookings available within the system.

---

## PUT Update Booking By Id

http://localhost:8070/Booking/update/65feba173526909ebd447f78

Users can update a specific booking via this API by using the booking Id.

### Body raw (json)

---

json

```
{
  "date": "2024.04.21",
  "timeDuration": "8:30 - 11:30",
  "description": "Projector for ITP",
  "faculty": "Engineering",
  "batchName": "Y2.S2.IT.WE.02.02",
  "bookingEntity": "PR567"
}
```

---

## DELETE Delete Booking By Id

http://localhost:8070/Booking/delete/65feba173526909ebd447f78

Users can delete a specific booking via this API by using the booking Id.

---

## GET View Booking By Id

http://localhost:8070/Booking/get/65feb9ba3526909ebd447f76

Users can view a specific booking via this API by using the booking Id.

---

## GET Search Booking By Keywords

<http://localhost:8070/Booking/search/Y3>

Users can view specific bookings via this API by using specific keywords regarding the booking.

---

## Time Table API Collection

This folder includes all the API regarding the time table management of the university. One can add new time tables, view all time tables, update time tables, delete time tables, view specific time tables and search specific needed time tables as per need using the API functions in this collection.

---

### POST Add Time Table

<http://localhost:8070/TimeTable/add>

This API allows users to create time tables for the university batches by filling out the respective details.

#### Body raw (json)

---

json

```
{
  "batch": "Y2.S2.DS.WE.01.01",
  "faculty": "Computing",
  "timeTableData": [
    {
      "date": "2024.03.24",
      "timeDuration": "09:00 AM - 11:00 AM",
      "course": "WKB",
      "location": "B502"
    },
    {
      "date": "2024.03.24",
      "timeDuration": "11:30 AM - 01:30 PM",
      "course": "MNX",
      "location": "A504"
    }
  ]
}
```

```
],
[
  {
    "date": "2024.03.25",
    "timeDuration": "02:00 PM - 04:00 PM",
    "course": "DS",
    "location": "G1103"
  },
  {
    "date": "2024.03.25",
    "timeDuration": "02:00 PM - 04:00 PM",
    "course": "ISDM",
    "location": "B503"
  }
]
]
```

---

## GET Get All Time Tables

<http://localhost:8070/TimeTable/>

Using this API users can view all the time tables available within the system.

---

## PUT Update Time Table By Id

<http://localhost:8070/TimeTable/update/65fed87c65b698d1f76c92e1>

Users can update a specific time table via this API by using the time table Id.

---

### Body raw (json)

json

```
{
  "batch": "Y2.S2.DS.WE.01.01",
  "faculty": "Computing",
  "timeTableData": [
```

```
[
  {
    "date": "2024.03.24",
    "timeDuration": "09:00 AM - 11:00 AM",
    "course": "BOP",
    "location": "B502"
  },
  {
    "date": "2024.03.24",
    "timeDuration": "11:30 AM - 01:30 PM",
    "course": "MNX",
    "location": "A504"
  }
],
[
  {
    "date": "2024.03.25",
    "timeDuration": "02:00 PM - 04:00 PM",
    "course": "DS",
    "location": "G1103"
  },
  {
    "date": "2024.03.25",
    "timeDuration": "02:00 PM - 04:00 PM",
    "course": "ISDM",
    "location": "B503"
  }
]
]
```

---

## DELETE Delete Time Table By Id

<http://localhost:8070/TimeTable/delete/65fed7b603ef78cbb22ca704>

Users can delete a specific time table via this API by using the time table Id.

---

## GET View Time Table By Id

<http://localhost:8070/TimeTable/get/65fed87c65b698d1f76c92e1>

Users can view a specific time table via this API by using the time table Id

Users can view a specific time table via this API by using the time table id.

---

## GET Search Time Table By Keywords

`http://localhost:8070/TimeTable/search/Y2`

Users can view specific time tables via this API by using specific keywords regarding the time table.

---

## Student API Collection

This folder includes all the API regarding the student management of the university. One can add new students, view all students, update students, delete students, view specific students, update student enrollment status and search specific needed students as per need using the API functions in this collection.

---

## POST Add Student

`http://localhost:8070/Student/add`

This API allows users to create student accounts for the university by filling out the respective details.

### Body raw (json)

---

json

```
{
  "registerNumber": "IT21341540",
  "studentName": "Senanayaka W.H.S.M",
  "studentAge": 23,
  "email": "heshan@gmail.com",
  "batchName": "Y3.S2.IT.WE.02.02",
  "faculty": "Computing",
  "specialization": "IT",
  "enrollmentStatus": true,
  "userType": "student",
```



```
"userName": "IT21340796",

"password": "hs123"
}
```

---

## GET Get All Students

<http://localhost:8070/Student/>

Using this API users can view all the students available within the system.

---

## PUT Update Student Acc. By Id

<http://localhost:8070/Student/update/65feed8d009b1057687bdb4e>

Users can update a specific student account via this API by using the student Id.

---

### Body raw (json)

json

```
{
  "registerNumber": "IT21338052",
  "studentName": "Pathirana J.K.H.",
  "studentAge": 23,
  "email": "keshan@gmail.com",
  "batchName": "Y3.S2.SE.WE.02.02",
  "faculty": "Computing",
  "specialization": "SE",
  "enrollmentStatus": true,
  "userType": "student",
  "userName": "IT21338052",
  "password": "kp123"
}
```

## **DELETE** Delete Student Acc. By Id

`http://localhost:8070/Student/delete/65feed06c2e48da4a0bf2dc2`

Users can delete a specific student account via this API by using the student Id.

---

## **GET** View Student Acc. By Id

`http://localhost:8070/Student/get/65feecc8c2e48da4a0bf2dc0`

Users can view a specific student account via this API by using the student Id.

---

## **GET** Search Student By Keywords

`http://localhost:8070/Student/search/IT21345296`

Users can view specific students via this API by using specific keywords regarding the student account.

---

## **PUT** Update Enrollment Status of Student By Id

`http://localhost:8070/Student/updateEnrollment/65feed8d009b1057687bdb4e`

Users can update enrollment status for the batch of the student via this API function using the student id.

**Body** raw (json)

---

json

```
{  
  "enrollmentStatus": false  
}
```

---

## Enrollment API Collection

This folder includes all the API regarding the **course enrollment management** of the students in the university. One can add new enrollments, view all enrollments, update enrollments, delete enrollments, view specific enrollments, update the enrollment status of the student for a specific course and search specific needed enrollments as per need using the API functions in this collection.

---

### POST Add Enrollment

<http://localhost:8070/Enrollment/add>

This API allows users to create course enrollment for the university by filling out the respective details.

#### Body raw (json)

---

json

```
{  
  "studentId": "IT21345296",  
  "faculty": "Computing",  
  "batchName": "Y3.S2.SE.WE.02.02",  
  "specialization": "SE",  
  "enrollmentStatus": true,  
  "course": "DSA"  
}
```

---

### DELETE Delete Enrollment By Id

http://localhost:8070/Enrollment/delete/65fef1b0677549920a680c69

Users can delete a specific course enrollment via this API by using the enrollment Id.

---

## **PUT** Update Enrollment Status By Id

http://localhost:8070/Enrollment/updateEnrollment/65fef1a9677549920a680c67

Users can update enrollment status for the course of the student via this API function using the enrollment id.

### **Body** raw (json)

---

**json**

```
{  
  "enrollmentStatus": false  
}
```