Authentication service

We are building a university management system where we have three types of roles such as Admin , Student , Faculty.

**Functional Requirements:**

* **Student**
  + Student can login and log out
  + Student can manage and update their profile.
  + Student can update certain fields
  + Student can enroll in a semester.
  + Student can enroll in offered courses for a specific semester.
  + Student can pay their tuition fees through offline or online.(Partial / Full Payment)
  + Student can see their transaction histories.
  + Student can see their class routines.
  + Student can see their otice board and events
  + Student can see their result (Full / Semester Wise).
  + Student can evaluate their teachers.
* **Admin:**
  + Admin can log in and log out
  + Admin can manage and update their profile.
  + Admin can only update certain fields.
  + Admin can manage user accounts
    - Block/Unblock
    - Change Password
    - Forcefully Log out
  + Admin can manage multiple process:
    - Semester
    - Offered Courses
    - Section
    - Faculty
    - Student
    - Building
    - Room
    - Payment
    - Permissions
    - Activity
* **Faculty:**
  + Faculty can log in and log out
  + Faculty can manage and update their profile.
  + Faculty can only update certain fields.
  + Faculty can manage user accounts
  + Faculty can manage student grades
  + Access to Academic and Personal Information.
  + Faculty can manage their lecture resources

**Models**

**Permission:**

* title

**UserPermission:**

* permissionId
* userId

**User:**

* id [ Generated ID , same as student id]
* role [ Append From Backend , user.service.ts]
* password [ default Password | , give password from frontend ]
* createdAt [mongoose by default]
* updatedAt [mongoose by default]
* student [reference ] (\_id mongoose )
* || admin [reference ] (\_id mongoose )
* || faculty [reference ] (\_id mongoose )

**Student:**

* id [ Generated ID, same as user id ]
* name
  + fisrtName
  + middleName
  + lastName
* Gender → enum → male || female
* dateOfBirth 24-04-1998
* email
* contactNo
* emergencyContactNo
* presentAddress
* permanentAddress
* bloodGroup → enum →
* guardian
  + fatherName
  + fatherOccupation
  + fatherContactNo
  + motherName
  + motherOccupation
  + motherContactNo
  + address
* localGuardian
  + name
  + occupation
  + contactNo
  + address
* academicSemester [ Reference]
* academicDepartment [Reference]
* academicFaculty [Reference]

**Sample Data:**

**{**

**"password": "123456",**

**"student": {**

**"name": {**

**"firstName": "Mezbaul",**

**"lastName": "Abedin",**

**"middleName": "Forhan"**

**},**

**"dateOfBirth": "24-04-1998",**

**"gender": "male",**

**"bloodGroup": "O+",**

**"email": "user@gmail.com",**

**"contactNo": "user\_4",**

**"emergencyContactNo": "01600000000",**

**"presentAddress": "CTG",**

**"permanentAddress": "CTG",**

**"academicFaculty": "6473b67a123df7493739e2a0",**

**"academicDepartment": "6473badee8ddae33f1cf0a3f",**

**"academicSemester": "647a5d967a96742c8cb4b8d2",**

**"guardian": {**

**"fatherName": "MD.ABBU",**

**"fatherOccupation": "Retired Teacher",**

**"fatherContactNo": "01600000000",**

**"motherName": "Mrs.Ammu",**

**"motherOccupation": "Housewife",**

**"motherContactNo": "01600000000",**

**"address": "CTG"**

**},**

**"localGuardian": {**

**"name": "Zahid Hasan",**

**"occupation": "Service Holder",**

**"contactNo": "01600000000",**

**"address": "Dhaka"**

**}**

**}**

**}**

**Faculty:**

* id (custom generated by generateFacultyId)
* name
  + firstName
  + midleName
  + lastName
* gender ('male'|'female')
* dateOfBirth
* email
* contactNo
* emergencyContactNo
* presentAddress
* permanentAddress
* bloodGroup ( 'A+' | 'A-' | 'B+' | 'B-' | 'AB+' | 'AB-' | 'O+' | 'O-')
* designation (Professor, Lecturer)
* academicDepartment (Department of Computer Science & Engineering) [reference]
* academicFaculty( Faculty of Science and Engineering ) [reference]

**Admin:**

* Id
* name
  + fisrtName
  + middleName
  + lastName
* gender
* dateOfbirth
* email
* contactNo
* emergencyContactNo
* department
* designation

**Academic Semester:**

* title
* year
* code
* startMonth
* endMonth

**Sample Data:**

* Autumn | Summer | Fall
* 01 | 02 | 03
* 2023
* January
* May

**Academic Faculty**

* title

**Sample Data:**

* Faculty of Science and Engineering
* Faculty of Business Administration
* Faculty of Arts and Social Science

1. Create Faculty
2. Read Faculty (Pagination)
3. Update faculty
4. Delete Faculty

**Academic Department**

* title
* academicFaculty (Reference)

**Sample Data:**

{

**title:** ‘Department of Computer Science’,

**academicFaculty:** 64820578aeaa15acb6483dc5

}

**After Populate:**

{

**title:** ‘Department of Computer Science’,

**academicFaculty:** {

**\_id**: 64820578aeaa15acb6483dc5,

**title**:’Faculty of Science and Engineering’

}

}

1. Create Faculty
2. Read Faculty (Pagination)
3. Update faculty
4. Delete Faculty

API END POINTS:

* **User**:
  + users/create-student (POST)
  + users/create-faculty (POST)
  + users/create-admin (POST)
  + users/my-profile (GET)
  + users/:id (GET)
  + users/:id (PATCH)
  + user/:id (DELETE)
  + users/:id/force-logged-out
  + users?page=1&limit=10 (GET)
  + users/:id/available-permissions?page=1&limit=10 (GET)
  + users/:id/assigned-permissions?page=1&limit=10 (GET)
  + users/:id/assign-permissions (POST)
  + users/:id/remove-permissions (POST)
* **Student:**
  + students (GET)
  + students?page=1&limit=10 (GET) (Pagination and Filtering)
  + students/:id (GET)
  + students/:id (PATCH)
  + students/:id (DELETE)
* **Faculty**:
  + faculties?page=1&limit=10&sortBy=gender&sortOrder=desc (GET)
  + faculties?searchTerm=lec (GET)
  + faculties/:id (GET)
  + faculties/:id (PATCH)
  + faculties/:id ( DELETE)
* **Admin**:
  + admins?page=1&limit=10 (GET)
  + admins/:id (GET)
  + admin/:id (PATCH)
* **Permission**
  + permissions?page=1&limit=10 (GET)
  + permissions (POST)
  + permissions/:id (GET)
  + permissions/:id (PATCH)
  + permissions/:id (DELETE)
* **Auth**
  + auth/login (POST)
    - id
    - password
  + auth/change-password (POST)
  + auth/refresh-token (POST)
  + auth/forgot-password(POST)
  + auth/reset-password (POST)
* **Academic Semester**
  + academic-semesters/create-semester (POST)
  + academic-semesters (GET) ( paginated)
  + academic-semesters/:id (GET)
  + academic-semesters/:id (PATCH)
  + academic-semesters/:id (DELETE)