***“PH University”***

***Requirement Analysis - MVP Project***

**MERN Stack MVP Web Application**

1. **Functional Requirements:**
   1. **Authentication:** 
      1. **Student:**
         1. Can login and logout securely.
         2. Can update their password.
      2. **Faculty:**
         1. Can login and logout securely.
         2. Can update their password.
      3. **Admin:**
         1. Can login and logout securely.
         2. Can update their password.
   2. **Profile Management:**
      1. **Student:**
         1. Can manage and update their profile information.
         2. Can update certain fields only.
      2. **Faculty:**
         1. Can manage and update their profile information.
         2. Can update certain fields only.
      3. **Admin:**
         1. Can manage and update their profile information.
         2. Can update certain fields only.
   3. **Academic Procedures:**
      1. **Student:** 
         1. Can enroll in offered courses for a specific semester.
         2. Can view their class schedule.
         3. Can view their grades and results.
         4. Can view notice boards and events.
      2. **Faculty:**
         1. Can manage students’ grades.
         2. Can access students’ personal and academic information.
      3. **Admin:**
         1. Can manage multiple processes:
            1. Semesters;
            2. Courses;
            3. Offered Courses;
            4. Sections;
            5. Rooms;
            6. Buildings;
   4. **User Management:**
      1. **Admin:**
         1. Can manage multiple types of users.
         2. Can block/unblock users.
         3. Can change users’ passwords.
2. **Data Model:**
   1. **Academic Semester:**
      1. \_id
      2. name
      3. year
      4. code
      5. startMonth
      6. endMonth
      7. createdAt
      8. updatedAt
   2. **Academic Faculty:**
      1. \_id
      2. name
      3. createdAt
      4. updatedAt
   3. **Academic Department:**
      1. \_id
      2. name
      3. academicFaculty
      4. createdAt
      5. updatedAt
   4. **Management Department:**
      1. \_id
      2. name
      3. createdAt
      4. updatedAt
   5. **Course :**
      1. \_id
      2. title
      3. prefix
      4. code
      5. credits
      6. prerequisiteCourse:
         1. course
         2. isDeleted
      7. isDeleted
      8. createdAt
      9. updatedAt
   6. **Semester Registration:**
      1. **Client Side Model:**
         1. academicSemester
         2. startDay
         3. endDay
         4. startTime
         5. endTime
         6. minCredit
         7. maxCredit
      2. **Database Model:**
         1. \_id
         2. academicSemester
         3. status
         4. startDate
         5. endDate
         6. minCredit
         7. maxCredit
         8. createdAt
         9. updatedAt
   7. **User:**
      1. \_id
      2. id (generated)
      3. password
      4. needsPasswordChange
      5. role
      6. status
      7. createdAt
      8. updatedAt
      9. isDeleted
   8. **Student:**
      1. \_id
      2. id (generated)
      3. user
      4. name
         1. firstName
         2. middleName
         3. lastName
      5. gender
      6. dateOfBirth
      7. email
      8. contactNo
      9. emergencyContactNo
      10. bloodGroup
      11. presentAddress
      12. permanentAddress
      13. guardian
          1. father
             1. name

firstName

middleName

lastName

* + - * 1. occupation
        2. contactNo
      1. mother
         1. name

firstName

middleName

lastName

* + - * 1. occupation
        2. contactNo
    1. localGuardian
       1. name
          1. firstName
          2. middleName
          3. lastName
       2. occupation
       3. contact No
       4. address
       5. relationship
    2. profileImage
    3. admissionSemester
    4. nationality
    5. Religion
    6. isDeleted
  1. **Faculty:**
     1. \_id
     2. id (generated)
     3. user
     4. designation
     5. name
        1. firstName
        2. middleName
        3. lastName
     6. gender
     7. dateOfBirth
     8. email
     9. contactNo
     10. emergencyContactNo
     11. bloodGroup
     12. presentAddress
     13. permanentAddress
     14. profileImage
     15. academicDepartment
     16. joiningDate
     17. nationality
     18. religion
     19. isDeleted
  2. **Admin:**
     1. \_id
     2. id (generated)
     3. user
     4. designation
     5. name
        1. firstName
        2. middleName
        3. lastName
     6. gender
     7. dateOfBirth
     8. email
     9. contactNo
     10. emergencyContactNo
     11. bloodGroup
     12. presentAddress
     13. permanentAddress
     14. profileImage
     15. managementDepartment
     16. joiningDate
     17. nationality
     18. religion
     19. isDeleted

1. **API Endpoints and CRUD Operation Methods:**
   1. **User:**
      1. **Create different types of users:**
         1. users/create-student (POST)
         2. users/create-faculty (POST)
         3. users/create-admin (POST)
   2. **Student:**
      1. **Get all students:** students (GET)
      2. **Get a specific student:** student/:id (GET)
      3. **Update a specific student:** student/:id (PATCH)
      4. **Delete a specific student:** student/:id (DELETE)
      5. **Get students’ profile:** students/my-profile (GET)
   3. **Faculty:**
      1. **Get all faculties:** faculties (GET)
      2. **Get a specific faculty:** faculty/:id (GET)
      3. **Update a specific faculty:** faculty/:id (PATCH)
      4. **Delete a specific faculty:** faculty/:id (DELETE)
      5. **Get faculties’** **profile:** faculties/my-profile (GET)
   4. **Admin:**
      1. **Get all admins:** admins (GET)
      2. **Get a specific admin:** admin/:id (GET)
      3. **Update a specific admin:** admin/:id (PATCH)
      4. **Delete a specific admin:** admin/:id (DELETE)
      5. **Get admins’ profile:** admins/my-profile (GET)
   5. **Auth:**
      1. **Login:** auth/login
      2. **Register:** auth/register
      3. **Refresh Token:** auth/refresh-token
      4. **Change Password:** auth/change-password
      5. **Forgot Password:** auth/forgot-password
      6. **Reset Password:** auth/reset-password