DDetailed Design Specification

1. Introduction
   * Background: This document is designed to provide an overview of the project. It includes the detailed design of the database, RESTful API Design, and frontend design.
2. Designed Design of the System
   * Database Tables
3. The relationship between **Application** and **Job** is **Many to One**, as a job can receive many applications. A foreign key to Job\_post\_id is stored in Application Table.
4. The relationship between **Application** and **Student** is **Many to One**, as asingle student can submit many applications. A foreign key to student \_id is stored in Application Table.
5. The relationship between **Job** and **Alumni** is **Many to One,** asa single alumni can post multiple jobs. A foreign key to Alumni \_id is stored in Job Table.

|  |  |
| --- | --- |
| User Table | |
| User\_id | Primary Key |
| Username | CharField (32) |
| Email | emailField |
| Password | charField (32) |
| Role | Enumerate (student/alumni) |
| First Name | CharField(32) |
| Last Name | CharField(32) |
| Location | CharField (255) |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

|  |  |
| --- | --- |
| Alumni Table | |
| Alumni\_id | Primary Key |
| User\_id | **Foreign Key references User Table** |
| Company\_name | CharField (255) |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

|  |  |
| --- | --- |
| Student Table | |
| Student\_id | Primary Key |
| User\_id | **Foreign Key references User Table** |
| School | CharField(32) |
| Year in School | PositiveIntegerField() |
| Major | CharField(32) |
| Graduation Year | DateTimeField() |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

|  |  |
| --- | --- |
| Job Posts Table | |
| Job\_post\_id | Primary Key |
| alumni\_id | **Foreign Key references Alumni Table (Many to one Relation)** |
| Job Name | CharField (64) |
| Job Company | CharField (64) |
| Job Requirement | TextField (255) |
| Job Description | TextField (255) |
| Status | Enum: (Accept/Closed) |
| **Question** | TextField (255) |
| **Num\_of\_applicants** | BigPositiveIntegerField |
| **Review\_process** | Enum: (in-review, pass, failed) |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

|  |  |
| --- | --- |
| Applications Table | |
| Application\_id | Primary Key |
| Status | Enum: (In progress/Selected/Not-Moving-Forward) |
| Student\_id | **Foreign Key references Student Table (Many to One Relation)** |
| Job\_post\_id | **Foreign Key references Job Posts Table (Many to One Relation)** |
| Resume | FileField |
| linkedin | URLField |
| application\_date | DateTimeField() |
| Answer | TextField (255) |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

Relation Tables: to store the many-to-many relationship between saved jobs and students.

|  |  |
| --- | --- |
| Favorite Job Table | |
| ID | Primary Key |
| Student\_id | Foreign Key References Student Table (Many to Many Relation) |
| Job\_id | Foreign Key References Job Table (Many to Many Relation) |
| Created\_time | DateTimeField |
| Modified\_time | DateTimeField |

* + Database View:
    - **CombineStudentInfo**: this contains all the information (user, student, application, favorite jobs) of a student. This will be used in the Student Profile Page.

CREATE VIEW CombineStudentInfo AS

SELECT

u.User\_id,

u.Email,

u.Role,

s.Student\_id,

s."First Name" AS Student\_First\_Name,

s."Last Name" AS Student\_Last\_Name,

s.School,

s."Year in School",

s.Major,

s.Location AS Student\_Location,

s."Graduation Year",

f.Student\_id AS Favorite\_Student\_id,

f.Job\_id AS Favorite\_Job\_id,

a.Application\_id AS Application\_id,

a.Status AS Application\_Status,

a.Job\_post\_id AS Application\_Job\_id

FROM

"User" u

LEFT JOIN

Student s ON u.User\_id = s.User\_id

LEFT JOIN

"Favorite Job Table" f ON s.Student\_id = f.Student\_id

LEFT JOIN

"Applications Table" a ON s.Student\_id = a.Student\_id;

* + - **CombineAlumniInfo**: this contains all the information (user, alumni, job posts) of an alumni. This will be used in the Alumni Profile page

CREATE VIEW CombineAlumniInfo AS

SELECT

u.User\_id,

u.Email,

u.Role,

a.Alumni\_id,

a.Company\_name,

a.Location AS Alumni\_Location,

j.Job\_post\_id,

j."Job Name" AS Job\_Name,

j."Job Requirement" AS Job\_Requirement,

j."Job Description" AS Job\_Description,

j.Status AS Job\_Status

FROM

"User" u

LEFT JOIN

Alumni a ON u.User\_id = a.User\_id

LEFT JOIN

"Job Posts Table" j ON a.Alumni\_id = j.alumni\_id

* + - **AlumniViewApplicationInfo**: this helps Alumni who reviews the applications from his job posts can say all the basic information of the student applicants.

CREATE VIEW "AlumniViewApplicationInfo" AS

SELECT

A."id" AS "application\_id",

A."status",

S."school" AS "student\_school",

S."major" AS "student\_major",

U."email" AS "student\_email",

U."first\_name" AS "student\_first\_name",

U."last\_name" AS "student\_last\_name",

AL."company\_name" AS "alumni\_company\_name",

AL."location" AS "alumni\_location"

FROM "Application" AS A

JOIN "Student" AS S ON A."student\_id" = S."id"

JOIN "User" AS U ON S."user\_id" = U."id"

JOIN "Alumni" AS AL ON A."alumni\_id" = AL."id";

1. Design RESTful API:
   * Authentication and Registration

* **POST /api/login/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/login /** | HTTP Method: | POST |
| Functionality | This endpoint is responsible for authenticating a user's login credentials and providing an authentication token upon successful login. |
| **@Parameters:** | |
| **email** (string) | The email address of the user. |
| **password** (string) | The password chosen by the user. |
| **@Returns :** | |
| **If Successful** | {  "success": true,  "message": "Login successful",  **‘return the basic user (with student/alumni) info; saved in cookie.’**  } |
| **If unsuccessful** | {  "success": false,  "message": "Login failed: Invalid email or password."  } |

.

* + Alumni Endpoints
    - **GET /api/alumni/**

1. **Return all alumni basic information.**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/** | HTTP Method: | GET |
| Functionality | This endpoint is designed to retrieve a comprehensive list of all alumni profiles available within the system. |
| **@Parameters:** | |
| None | None |
| **@Returns** | |
| **If Successful** | {  "success": true,  "alumni": [  {  "alumni\_id": 1,  "first\_name": "John",  "last\_name": "Doe",  "email": "john.doe@example.com",  "company\_name": "TechCorp",  },  }  // ... other alumni profiles ...  ]  } |
| **If unsuccessful** | {  "success": false,  "error": "Unable to fetch alumni profiles. Please try again later."  } |

* + - **GET /api/alumni/<int:alumni\_id>/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>** | HTTP Method: | GET |
| Functionality | This endpoint is designed to retrieve details of a specific alumni profile based on the provided **alumni\_id**. |
| **@Parameters:** | |
| **alumni\_id** (int) | The unique identifier of the alumni profile being requested. |
| **@Return** | Details of **the specific alumni profile**, including information such as alumni\_id, first\_name, last\_name, email, company\_name, and potentially **associated job post**s. |
| **Sample Successful Return** | {  "success": true,  "alumni": {  "alumni\_id": 1,  "first\_name": "John",  "last\_name": "Doe",  "email": "john.doe@example.com",  "company\_name": "Google",  "job\_posts": [  {  "job\_post\_id": 101,  "job\_name": "Software Engineer",  "job\_requirement": "Bachelor's degree in CS",  "job\_description": "Develop and maintain...",  "status": "Accept"  },  // ... other job posts  ]  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Alumni profile with ID 123 not found."  } |

* + - **POST /api/alumni/**

1. **Register User profile first and get the User\_id**
2. **Then, use that User\_id and rest of information to create alumni profile.**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>** | HTTP Method: | POST |
| Functionality | This endpoint is responsible for creating a new alumni profile after user registration. |
| **@Parameters:** | |
| **Email** string) | The email of the user. |
| **username** (string) | The username of the profile |
| **password** (string) | The password chosen by the user. |
| **role** (string) | The role of the user (either 'user' or 'admin'.). |
| **location (string)** | The location of the user. |
| **user\_id** (integer) | The unique identifier of the user associated with the new alumni profile. (get this id after user profile is created) |
| **company\_name** (string) | The name of the company associated with the alumni profile. |
| **email** (string) | The email address of the user. |
| **first\_name (string)** | The first name of the user. |
| **last\_name** (string) | The last name of the user. |

* + - **PUT /api/alumni/<int:alumni\_id>/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>** | HTTP Method: | PUT |
| Functionality | This endpoint allows updating an existing alumni profile's Company Name. |
| **@Parameters:** | |
| **alumni\_id** (integer) | The unique identifier of the alumni profile being updated. |
| **company\_name** (string) | The updated name of the company associated with the alumni profile |
| **@Return** | The newly created alumni profile. |
| **Sample Successful Return** | {  "success": true,  "alumni": {  "alumni\_id": 123,  "user\_id": 456,  "company\_name": "NewTech",  "location": "San Francisco, CA"  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to update alumni profile. The specified alumni ID does not exist."  } |

* + - **DELETE /api/alumni/<int:alumni\_id>/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>/** | HTTP Method: | DELETE |
| Functionality | Delete the specific alumni profile based on the provided **alumni\_id.** |
| **@Parameters:** | |
| **alumni\_id** (integer) | The unique identifier of the alumni profile to be deleted. |
| **@Return** | Notification if the delete is successful or not successful. |
| **Sample Successful Return** | {  "success": true,  "message": "Alumni profile with ID 123 has been deleted."  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to delete alumni profile. The specified alumni ID does not exist."  } |

* + Student Endpoints
    - **GET /api/students/**

1. **Get All information for STUDENT (role: administrator).**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/students/** | HTTP Method: | GET |
| Functionality | This endpoint lists all student profiles along with their application information. |
| **@Parameters:** | |
| None | The unique identifier of the alumni profile to be deleted. |
| **@Return** | A response containing a list of all student profiles. Each student profile includes details such as **student\_id**, **user\_id**, **school**, **year\_in\_school**, **major**, **location**, **graduation\_year**, and information about their applications. |
| **Sample Successful Return** | {  "success": true,  "students": [  {  "student\_id": 1,  "user\_id": 101,  "school": "University of Tech",  "year\_in\_school": 3,  "major": "Computer Science",  "location": "Cityville",  "graduation\_year": "2023-05-15T00:00:00Z",  "created\_date": "2023-08-27T00:00:00Z",  "modified\_date": "2023-08-27T00:00:00Z"  }  ]  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to retrieve student profiles. Please try again later."  } |

* + - **GET /api/students/<int:student\_id>/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/students/<int:student\_id>** | HTTP Method: | GET |
| Functionality | This endpoint retrieves details of a specific student profile based on the provided **student\_id**. |
| **@Parameters:** | |
| **student\_id** (integer): | The unique identifier of the student profile to be requested. |
| **@Return** | Details of the specific student profile, including information such as **student\_id**, **user\_id**, **school**, **year\_in\_school**, **major**, **location**, **graduation\_year**, and potentially associated application information. |
| **Sample Successful Return** | {  "success": true,  "student": {  "student\_id": 1,  "user\_id": 101,  "school": "University of Tech",  "year\_in\_school": 3,  "major": "Computer Science",  "location": "Cityville",  "graduation\_year": "2023-05-15T00:00:00Z",  "applications": [  {  "application\_id": 201,  "status": "In progress",  "job\_post\_id": 101,  "resume": "link\_to\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/student1",  "application\_date": "2023-08-24T10:00:00Z"  },  // ... other applications associated with this student ...  ]  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Student profile with ID 123 not found."  } |

* + - **POST /api/students/** 
      1. **Register User profile first and get the User\_id**
      2. **Then, use that User\_id and rest of information to create student profile.**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/students/<int:student\_id>** | HTTP Method: | POST |
| Functionality | This endpoint is responsible for creating a new student profile after user registration. |
| **@Parameters:** | |
| **Email** string) | The email of the user. |
| **username** (string) | The username of the profile |
| **password** (string) | The password chosen by the user. |
| **role** (string) | The role of the user (either 'user' or 'admin'.). |
| **location** (string) | The location of the user. |
| **user\_id** (integer) | The unique identifier of the user associated with the new alumni profile. (Get this id after student profile is created) |
| **year\_in\_school**(integer): | The year in school that the student is currently in. |
| **major** (string): | The major or field of study of the student. |
| **graduation\_year** (datetime) | The expected graduation year of the student. |
| **@Return** | The newly created student profile. |
| **Sample Successful Return** | {  "success": true,  "student": {  "student\_id": 123,  "user\_id": 456,  "school": "University of Tech",  "year\_in\_school": 3,  "major": "Computer Science",  "graduation\_year": "2023-05-15T00:00:00Z"  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to create student profile. The specified student ID does not exist."  } |

* + - **PUT /api/students/<int:student\_id>/**
      1. Frontend will restrict **only the modified parameter** will be sent in the request.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/students/<int:student\_id>/** | HTTP Method: | PUT |
| Functionality | This endpoint allows updating an existing student profile's information. |
| **@Parameters:** | |
| **student\_id** (integer): | The unique identifier of the student profile being updated. |
| **school** (string, optional): | The updated name of the school attended by the student. |
| **year\_in\_school**(integer, optional): | The updated year in school that the student is currently in. |
| **major** (string, optional): | The updated major or field of study of the student. |
| **graduation\_year** (datetime, optional) | The updated expected graduation year of the student. |
| **@Return** | The updated student profile. |
| **Sample Successful Return** | {  "success": true,  "student": {  "student\_id": 123,  "user\_id": 456,  "school": "Updated University",  "year\_in\_school": 4,  "major": "Updated Computer Science",  "graduation\_year": "2024-05-15T00:00:00Z"  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to update student profile. The specified student ID does not exist."  } |

* + - **DELETE /api/students/<int:student\_id>/**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/students/<int:student\_id>/** | HTTP Method: | DELETE |
| Functionality | This endpoint allows deleting a specific student profile based on the provided **student\_id**. |
| **@Parameters:** | |
| **student\_id** (integer) | The unique identifier of the student profile to be deleted. |
| **@Return** | Notification if the delete is successful or not successful. |
| **Sample Successful Return** | {  "success": true,  "message": "Student profile with ID 123 has been deleted."  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Failed to delete student profile. The specified alumni ID does not exist."  } |

* + Job Post Endpoints
    - **GET /api/jobposts/**
      1. **For administrator: see all jobs.**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/jobposts/** | HTTP Method: | GET |
| Functionality | Retrieves a list of all job posts. |
| **@Parameters:** | |
| N/A | N/A |
| **@Return** | List of job posts, each containing job details. |
| **Sample Successful Return** | [  {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_requirement": "Bachelor's in Computer Science",  ‘job\_company’ : “GOOGLE”  "job\_description": "Develop and maintain software applications.",  "status": "Accept"  },  // ... other job posts ...  ] |
| **Unsuccessful Return** | {  "error": "Failed to retrieve job posts",  "message": "An error occurred while trying to fetch job posts from the database."  } |

* + - **GET /api/jobposts/<int:job\_post\_id>/**
      1. If this review\_processis **fail** or **in-review,** then return 404.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/jobposts/** | HTTP Method: | GET |
| Functionality | Retrieve details of a specific job post. |
| **@Parameters:** | |
| **job\_post\_id** (integer) | The ID of the job post to retrieve. |
| **@Return** | Details of the specific job post. |
| **Sample Successful Return** | {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_requirement": "Bachelor's in Computer Science",  "job\_description": "Develop and maintain software applications.",  "status": "Accept"  } |
| **Unsuccessful Return** | {  "error": "Job post not found",  "message": "The specified job post ID does not exist in the database.”  } |

* + - **POST /api/jobposts/** 
      1. Set review\_process to be ‘in-review’

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/jobposts/** | HTTP Method: | POST |
| Functionality | Create a new job post by an alumni. |
| **@Parameters:** | |
| **alumni\_id** (integer) | The ID of the alumni creating the job post. |
| **job\_name** (string) | The name of the job |
| **location** (string) | The location of the job. |
| **job\_requirement** (string) | The requirements for the job |
| **Status** (string) | The status of the job post (e.g., "Accept" or "Closed"). |
| **@Return** | Details of the newly created job post, with num\_of\_application set to 0. |
| **Sample Successful Return** | {  "success": true,  "data": {  "job\_post\_id": 42,  "alumni\_id": 123,  "job\_name": "Software Developer",  "location": "New York",  "job\_requirement": "Bachelor's in Computer Science",  "job\_description": "Develop software solutions.",  "status": "Accept"  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Invalid input",  "message": "Please provide valid values for all required parameters."  } |

* + - **PUT /api/jobposts/<int:job\_post\_id>/**
      1. Update a job post (by that specific alumni)
      2. Parameters: **job\_name, location, job\_requirement, job\_description, status**
      3. **Authentication**: Requires authentication as an alumni user.
      4. **Authorization**: The authenticated user must be the original alumni creator of the job post.
      5. Returns: Success message or updated job post if authorized

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/jobposts/<int:job\_post\_id>/** | HTTP Method: | PUT |
| Functionality | Update a job post by the alumni who created it. |
| **@Parameters:** | |
| **Job\_company** (string) | The ID of the alumni creating the job post. |
| **job\_name** (string) | The name of the job |
| **location** (string) | The location of the job. |
| **job\_requirement** (string) | The requirements for the job |
| **Status** (string) | The status of the job post (e.g., "Accept" or "Closed"). |
| **@Return** | Details of the updated job post. |
| **Sample Successful Return** | {  "application\_id": 42,  "status": "In progress",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "Jane",  "last\_name": "Smith",  "major": "Computer Science",  "contact\_info": "jane@example.com",  "school\_year": 3  },  "job\_post": {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_company": "TechCorp"  }  } |
| **Unsuccessful Return** | {  "success": false,  "error": "Unauthorized",  "message": "You are not authorized to update this job post."  } |

**DELETE /api/jobposts/<int:job\_post\_id>/**

1. Delete a job post (by that specific alumni)
2. Returns: Success message or Error message

|  |  |  |
| --- | --- | --- |
| End Point:  /api/jobposts/<int:job\_post\_id>/ | HTTP Method: | DELETE |
| Functionality | Delete a job post (by an alumnus who created the job post). |
| @Parameters: | |
| **Job\_post\_id** (integer) | The id of the Job Post to be deleted |
| **@Return** | Message indicating the successful deletion of job posts |
| **Sample Successful Return** | {  “success:” True,  "message": "Job post deleted successfully"  } |
| **Unsuccessful Return** | {  “success:” False,  "message": "Job post not found"  } |

* + Application Endpoints
    - **GET /api/applications/**

1. Retrieve a list of all applications (for administrator).
2. This endpoint is used to fetch and display a comprehensive list of all the applications that have been submitted for various job posts. It provides an overview of application data in the system.
3. Returns: A JSON array containing objects representing each application. Each application object includes information such as **application ID, status, applicant details, job post details.**

|  |  |  |
| --- | --- | --- |
| End Point:  /api/applications/ | HTTP Method: | GET |
| Functionality | Retrieve a list of all applications. |
| @Parameters: | |
| **N/A** | N/A |
| **@Return** | fetch and display all the applications that have been submitted for various job posts. |
| **Sample Successful Return** | [  {  "application\_id": 1,  "status": "In progress",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "John",  "last\_name": "Doe"  },  "job\_post": {  "job\_post\_id": 42,  "job\_name": "Software Developer",  "job\_company": "TechCorp"  },  "resume": "/media/resumes/john\_doe\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/johndoe",  "application\_date": "2023-08-25T12:34:56Z"  },  // ... other applications ...  ] |
| **Unsuccessful Return** | {  "message": "An error occurred while trying to fetch applications."  } |

* + - **GET /api/applications/<int:application\_id>/**

1. Retrieve a specific application for that student who apply for this role.
2. Returns: Details of the specific application

|  |  |  |
| --- | --- | --- |
| End Point:  /api/applications/<int:application\_id> | Http Method: | GET |
| Functionality | Retrieve a specific application for a student who applied for a job role. |
| @Parameters | |
| **N/A** | N/A |
| **@Return** | Details of the specific application, including the application status, student details, related job post, and application-related information |
| **Sample Successful Return** | {  "application\_id": 1,  "status": "In progress",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "John",  "last\_name": "Doe"  }, "resume": "/media/resumes/john\_doe\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/johndoe",  "application\_date": "2023-08-25T12:34:56Z",  ‘question’ : ‘xxxx’,  ‘modified time’ : ‘2023-08-25T12:34:56Z’,  ‘created\_time’ : ‘2023-08-25T12:34:56Z’  } |
| **Unsuccessful Return** | {  "message": "Application not found."  } |

* + - **POST /api/applications/<int:student\_id>**

1. Apply for a job post (by student)
2. Parameters: **status, student\_id, job\_post\_id, resume, linkedin, application\_date**
3. This endpoint allows a student to submit an application for a specific job post (this job’s review\_process must be ‘pass’).
4. The num\_of\_applicant associated with this job will be added by 1 for each application.
5. Returns: Newly created application

|  |  |  |
| --- | --- | --- |
| End Point:  **/api/applications/<int:student\_id>** | Http Method: | POST |
| Functionality | Apply for a job post by a student. |
| @Parameters: | |
| **status** (string) | The status of the application (e.g., "In progress"). |
| **student\_id** (integer) | The ID of the student applying for the job. |
| **job\_post\_id** (integer) | The ID of the job post for which the student is applying. |
| **resume** (pdf) | The uploaded resume file. |
| **linkedin** (string) | The student's LinkedIn profile URL. |
| **application\_date** (string) | The date and time when the application is submitted. |
| **@Return** | Details of the newly created application, including application ID, status, student details, job post details, and application-related information. |
| **Sample Successful Return** | {  "application\_id": 42,  "status": "In progress",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "Jane",  "last\_name": "Smith"  },  "job\_post": {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_company": "TechCorp"  },  "resume": "/media/resumes/jane\_smith\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/janesmith",  "application\_date": "2023-08-25T12:34:56Z"  } |
| **Unsuccessful Return** | {  "message": "Failed to create application. Please provide valid values."  } |

* + - **PUT /api/applications/<int:application\_id>/**

1. Update an application (**by student**)
2. **Authentication: that student who submitted the application.**
3. **Condition 1: that job must not be reviewed by the alumni. (application\_status == ‘in progress’)**
4. **Condition 2: that job post must be submitted within 1 day. (job\_post.created\_date - created\_date <= 1)**
5. Parameters**: resume, linkedin**
6. Returns: Updated application

|  |  |  |
| --- | --- | --- |
| End Point:  **/api/applications/<int:student\_id>** | HTTP Method: | PUT |
| Functionality | Apply for a job post by a student. |
| @Parameters: | |
| **resume** (pdf) | The uploaded resume file. |
| **linkedin** (string) | The student's LinkedIn profile URL. |
| **application\_date** (string) | The date and time when the application is submitted. |
| **@Return** | Updated details of the application |
| **Sample Successful Return** | {  "application\_id": 42,  "status": "In progress",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "Jane",  "last\_name": "Smith"  },  "job\_post": {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_company": "TechCorp"  },  "resume": "/media/resumes/jane\_smith\_updated\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/janesmith\_updated",  "application\_date": "2023-08-25T12:34:56Z"  } |
| **Unsuccessful Return** | {  "message": "You are not authorized to update this application."  } |

* + - **GET /api/applications/<int:application\_id>**

1. View an application (by alumni)
2. **Authentication: the alumni who post this job**
3. Returns: Detailed information about the application, **including student major, student contact-info, student schoolyear, job post details, and application status.**

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>/status** | HTTP Method: | GET |
| Functionality | This endpoint allows an alumnus to view detailed information about a specific application, including the student's major, contact information, school year, and application status. |
| **@Parameters:** | |
| **status** (string) | The updated status of the application (e.g., "Selected"). |
| **application\_id** (string) | The ID of the application that is viewed by teh |
| **@Return** | Details of the updated application, including the updated status and other application-related information |
| **Sample Successful Return** | {  "application\_id": 42,  "status": "Selected",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "Jane",  "last\_name": "Smith"  },  "resume": "/media/resumes/jane\_smith\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/janesmith",  "application\_date": "2023-08-25T12:34:56Z"  } |
| **Unsuccessful Return** | {  "message": "Failed to update application status."  } |

* + - **PUT /api/applications/<int:application\_id>/**

1. Change the status of an application (by alumni)
2. **Authentication: the alumni who post this job**
3. Parameters: status
4. This endpoint allows alumni to update the status of an application (e.g., from "In progress" to "Selected" or “Not Moving Forward”).
5. Returns: Success message if the status change is successful, or an error message if the status could not be updated.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/alumni/<int:alumni\_id>/status** | HTTP Method: | PUT |
| Functionality | This endpoint allows alumni to update the status of a specific application. |
| **@Parameters:** | |
| **status** (string) | The updated status of the application (e.g., "Selected"). |
| **application\_id** (string) | The ID of the application that is viewed |
| **@Return** | Details of the updated application, including the updated status and other application-related information |
| **Sample Successful Return** | {  "application\_id": 42,  "status": "**Selected**",  "student": {  "student\_id": 123,  "user\_id": 456,  "first\_name": "Jane",  "last\_name": "Smith"  },  "job\_post": {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_company": "TechCorp"  },  "resume": "/media/resumes/jane\_smith\_resume.pdf",  "linkedin": "https://www.linkedin.com/in/janesmith",  "application\_date": "2023-08-25T12:34:56Z"  } |
| **Unsuccessful Return** | {  "message": "Failed to update application status."  } |

* + Favorite Job
    - **POST /api/favorite-jobs-saving/**

1. Save a job as a favorite (by student)
2. Parameters: student\_id, job\_post\_id.
3. This endpoint allows students to save a job as one of their favorite jobs.
4. Returns: Success message or error message.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/favorite-jobs-saving/** | HTTP Method: | POST |
| Functionality | Save a job as a favorite by a student. |
| **@Parameters:** | |
| **Job\_post\_id** (integer) | The ID of the job that to be saved as favorite |
| **@Return** | Success message indicating that the job post has been saved as a favorite. |
| **Sample Successful Return** | {  "message": "Job post saved as a favorite"  } |
| **Unsuccessful Return** | {  "message": "Failed to save this job into favorite."  } |

* **GET /api/favorite-jobs-display/<int:student\_id>/**

1. Retrieve a list of favorite jobs for a student.
2. This endpoint allows students to retrieve a list of job posts that they have saved as favorites.
3. Returns: List of favorite job posts.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/favorite-jobs-display/<int:student\_id>/** | HTTP Method: | GET |
| Functionality | Retrieve a list of favorite jobs for a student. |
| **@Parameters:** | |
| **student\_id** (integer) | The ID of the student for whom to retrieve favorite jobs |
| **@Return** | A list of job posts that the student has saved as favorites |
| **Sample Successful Return** | [  {  "job\_post\_id": 1,  "job\_name": "Software Engineer",  "job\_company": "TechCorp"  },  {  "job\_post\_id": 2,  "job\_name": "Data Analyst",  "job\_company": "DataTech"  }  ] |
| **Unsuccessful Return** | [] |

* **DELETE /api/favorite-jobs-delete/<int:favorite\_id>/**

1. Remove a job from favorites (by student)
2. This endpoint allows students to remove a job from their list of favorite jobs.
3. Returns: Success message or error message.

|  |  |  |
| --- | --- | --- |
| **End Point:**  **/api/favorite-jobs-delete/<int:student\_id>/** | HTTP Method: | DELETE |
| Functionality | Remove a job from favorites by a student. |
| **@Parameters:** | |
| **fa\_id** (integer) | The ID of the student for whom to retrieve favorite jobs |
| **@Return** | Remove a job from favorites by a student. |
| **Sample Successful Return** | {  "message": "Job post removed from favorites"  } |
| **Unsuccessful Return** | {  "message": "Favorite job entry not found"  } |