

GET -

/posting

```
SELECT job_name, posting_id, company_id, location, post_date  
FROM posting;
```

/posting/:id:

```
SELECT *  
FROM posting  
WHERE posting_id = :id;;
```

/user/:id:/applications

```
SELECT posting_id  
FROM applications  
WHERE user_id = :id;;
```

/company/:id:

```
SELECT * FROM employer_companies  
WHERE company_id = :id;;
```

/user/auth/:id:

```
SELECT * FROM user  
WHERE authuid = :id;;
```

/company

```
SELECT * FROM employer_companies;
```

/company/:id:/postings

```
SELECT * FROM posting  
WHERE company_id = :id;;
```

POST -

/user/create

POST

```
{
    school_id: Int?,
    company_id: Int?,
    year: Int,
    user_name: String,
    skills: String,
    authuid: String
}
```

.....

/application/create

POST

```
{
    posting_id: Int,
    user_id: Int
}
```

INSERT INTO applications

VALUES (:posting_id:, :user_id:, NOW());

/

posting/create

```
POST {
    posting_id: Int,
    job_name: String,
    job_description: String,
    med_salary: Double,
    sponsor: Int,
    remote_allowed: String,
    location: String,
    ng_or_internship: String,
    company_id: Int
}
```

```
INSERT INTO posting
VALUES (
    :posting_id:,
    :job_name:,
    :job_description:,
    :med_salary:,
    :sponsor:,
    :remote_allowed:,
    :location:,
    NOW(),
    :ng_or_internship:,
    :company_id:
);
```

UPDATE -

/user/:id:/update

```
POST {
    school_id: Int,
    company_id: Int,
    year: Int,
    user_name: String,
    skills: String,
    authuid: String
}
```

```
UPDATE user
```

```
SET
```

```
    school_id = :school_id:,
    company_id = :company_id:,
    year = :year:
    user_name = :user_name:
    skills = :skills:
```

```
WHERE user_id = :id;;
```

/posting/:id:/update

```
POST {
```

```

    job_name: String?,
    job_description: String?,
    med_salary: Int?,
    sponsor: Int?,
    remote_allowed: Int?,
    location: String?,
    ng_or_internship: String?,
    company_id: Int?
}

```

UPDATE posting

SET

```

    job_name = :job_name:,
    job_description = :job_description:,
    med_salary = :med_salary:,
    sponsor = :sponsor:,
    remote_allowed = :remote_allowed:,
    location = :location:,
    ng_or_internship = :ng_or_internship:,
    company_id = :company_id:

```

WHERE posting_id = :id;;

/application/:user_id/:posting_id:/update

POST {

status: String?

}

UPDATE applications

SET status = :status:

WHERE posting_id = :posting_id: AND user_id = :user_id;;

DELETE - (!!!! need triggers to remove dependent rows)

/posting/:id:/delete

DELETE FROM posting

```
WHERE posting_id = :id;;
```

/application/:user_id/:posting_id:/delete

```
DELETE FROM applications
```

```
WHERE posting_id = :posting_id: AND user_id = :user_id;;
```

/user/:id:/delete

```
DELETE FROM user
```

```
WHERE user_id = :id;;
```

TRIGGER -

```
CREATE TRIGGER update_user_streak_score  
AFTER INSERT ON applications  
FOR EACH ROW  
BEGIN
```

```
    DECLARE last_app_date DATETIME;
```

```
    DECLARE streak INT;
```

```
    DECLARE new_score DOUBLE;
```

```
    DECLARE time_diff INT;
```

```
    SELECT last_app_date FROM user WHERE user_id = NEW.user_id  
    INTO last_app_date;
```

```
    UPDATE user SET last_app_date = NEW.application_date WHERE user_id = NEW.user_id;
```

```
    SELECT current_streak INTO streak FROM user WHERE user_id = NEW.user_id;
```

```
        SELECT DATEDIFF(last_app_date, NOW()) INTO time_diff;
```

```
    IF last_app_date IS NULL OR time_diff >= 3 THEN
```

```
        SET streak = 0;
```

```
    ELSE
```

```
        IF time_diff >= 1 AND streak <= 7 THEN
```

```
            SET streak = streak + 1;
```

```
        END IF;
```

```
    END IF;
```

```
    UPDATE user
```

```
    SET current_streak = streak
```

```
    WHERE user_id = NEW.user_id;
```

```
SET new_score = 10 * POW(1.2, streak);

UPDATE user
SET points = points + new_score
WHERE user_id = NEW.user_id;
END;
```

PROCEDURE -

```
CREATE PROCEDURE GenerateLeaderboard()
BEGIN
    CREATE TEMPORARY TABLE TempLeaderboard AS
    SELECT
        s.school_id AS school_id,
        s.school_name AS school_name,
        SUM(points) AS total_score,
        SUBSTRING_INDEX(GROUP_CONCAT(user_name ORDER BY points DESC), ',', 4) AS
student_names
    FROM
        user AS u
        LEFT JOIN school AS s ON u.school_id = s.school_id
    GROUP BY
        school_id, school_name
        HAVING school_id IS NOT NULL
    LIMIT 50;

    DELETE FROM leaderboard;

    INSERT INTO leaderboard (school_id, school_name, total_score, student_names)
    SELECT school_id, school_name, total_score, student_names
    FROM TempLeaderboard;

    DROP TEMPORARY TABLE TempLeaderboard;
END;

CREATE EVENT IF NOT EXISTS RunLeaderboardUpdate
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_DATE + INTERVAL 1 DAY
DO
CALL GenerateLeaderboard();
```