For each endpoint write up the following based on a local copy of your database using fake data (with at least a million rows):

- **A profile of the endpoints using EXPLAIN before any indexes are added**
- **The SQL on relevant indexes that will impact the performance of that endpoint and WHY those indexes are the right indexes**
- **A profile of the endpoints using EXPLAIN after all indexes are added**

POST /users/

- Due to this being a post, there isn't rally an index that could improve this endpoint since there is no querying the database

GET /users/{user_id}

CREATE INDEX ON users (id)

- This will improve the peformance of this endpoint as it queries for the specific user based on their id. Therefore, the executer won't have to traverse the whole page in a sequential scan and can rather use the indexing of the id

POST /users/login

CREATE INDEX ON users(name)

- In this endpoint, a query is run to find the user with the associated name, therefore by adding an index for the users table basd on the name, there will be a faster lookup rather than a full table scan

GET /workouts/{user_id}

CREATE INDEX ON workouts(user_id)

- This will improve the performance of this endpoint as the associated workout will be indexed via the user id, where as before the endpoint would have to traverse the entire page of workouts to find the right workout, but now that is not an issue as the index will provide a reference to the proper workout

GET /user/{user_id}/logs

CREATE INDEX ON logs(user_id)

- This will improve the performance of this endpoint as it queries the logs table to find the logs with the same user_id as the one passed in. As a result, the executer won't have to do a full scan of the logs table and will be able to look up the logs based on the

POST /user/{user_id}/logs

- This endpoint would not benefit from any indexes as the only relevant index would to be to index the user table based on id to find the user, but this won't provide a large performance impact as the id is the users table's primary key

POST /goals/

- Much like the previous endpoint, there is only a query to search for the associated user, which is handled by the primary key of the id

POST /user/{user_id}/projection

CREATE INDEX ON logs(user_id)

- This will improve the performance of this endpoint as it queries the logs table to find the logs with the same user_id as the one passed in. As a result, the executer won't have to do a full scan of the logs table and will be able to look up the logs based on the

GET /user/{user_id}/projection

CREATE INDEX ON projection(user_id)

- This will improve the performance of this endpoint as the executuro previouslyt had to do a full table scan of the projection table in order to find all the projections with the same user_id. This will no longer occur as the indexes will have pointers to the locations projections which match the user_id