DBMS-Application

How the application was built

The application development process involved downloading the Software Engineering dataset from the Stack Exchange Archive. The downloaded XML files were parsed using a Python script. Then, `init.sql` was created using a pre-existing DDL, and various integrity constraints were added. The data was inserted using `data.sql`.

After the database was created, the backend was developed by creating various controllers, models and routes. The frontend was created in parallel using NextJS. Authentication was added along with the middleware after this, and the frontend was finally integrated with the backend.

Overall system architecture

- The `backend` folder consists of `controllers`, `db`, `middleware`, and `routes`, in addition to files like `./index.js` and `models.js`.
  - `controllers` has several files for the various backend APIs/functions. Examples of these include the APIs for creating answers (in `answer.js`), for signing in (in `auth.js`) and for autocomplete (in `autocomplete.js`).
  - `db` has various database related files. `init.sql` defines the DDL, `data.sql` inserts data into the created tables, and `drop.sql` drops the created tables. The problem-statement also asked for accounts to be created for pre-existing users, with a username and the default password as the username. For this, `gen_cred.py` populates the *credentials* table with usernames (display_name concatenated with user ID) and passwords (same as username).
-

Programming Languages used for different components

Group Member Contributions

- Kushagra Gupta -
- Rahul Ramachandran -
- Rishit D -
- Suryaansh Jain -