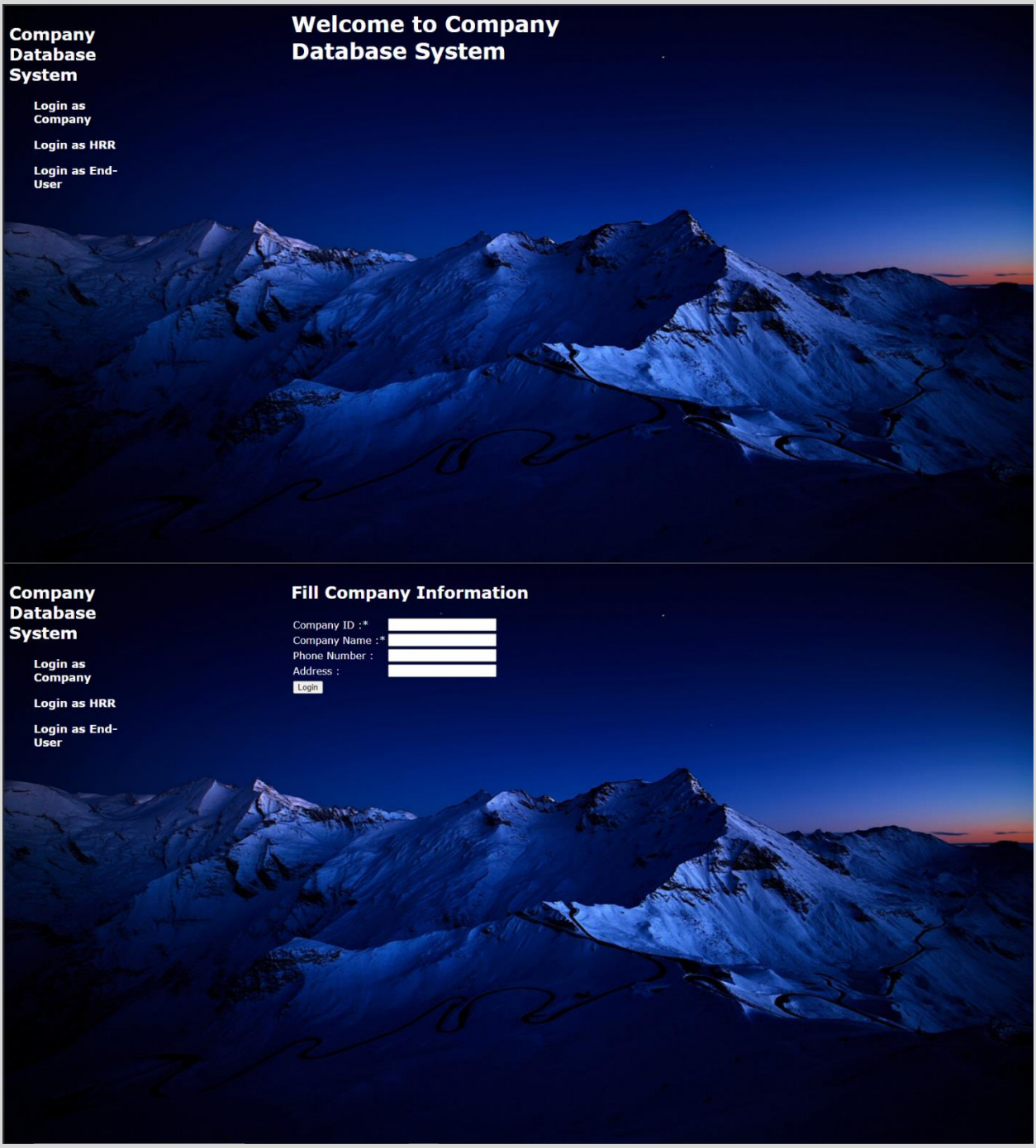
COMP2222 PROJECT 3 REPORT

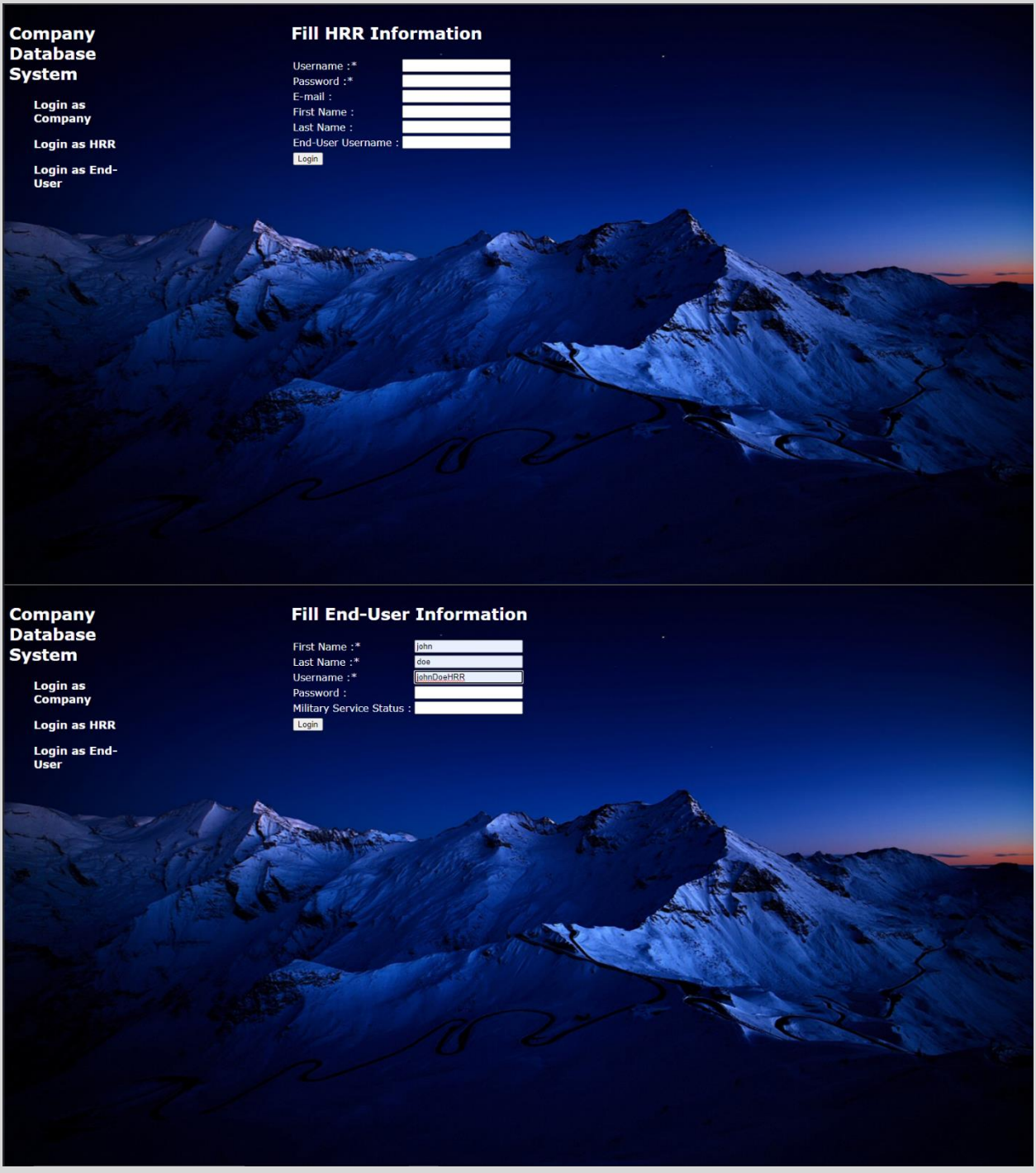
Design a database for a job postings application and design a graphical user interface that will connect to the database remotely to provide the services to an end-user.

PREPARED BY: **İlkim Sevinç**
Rana Selim

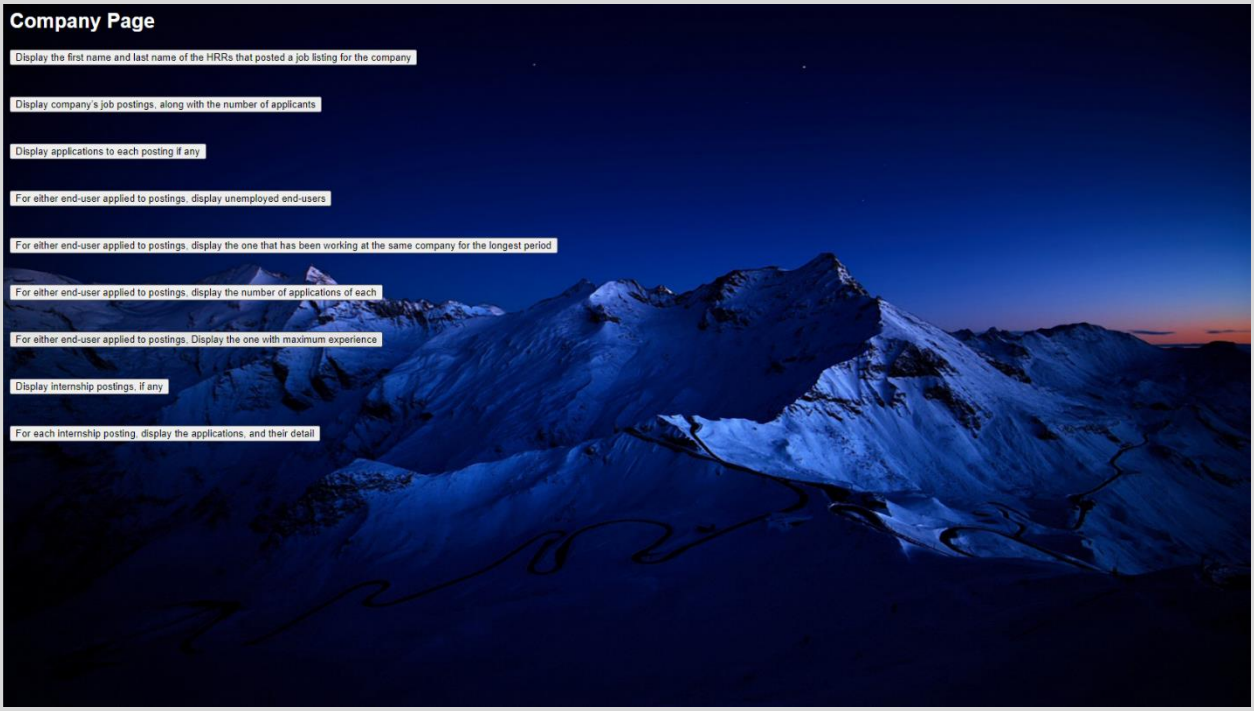
1- How is the GUI used?

First, we looked at what GUI means and what it is supposed to look like. With knowing ISO 9241 Standart we thought about the most efficient way to design it. We researched how a human reacts when they are using an app. We changed places and fonts of some text in order to make it easier to use this program. Our program is mostly blue and pastel color since it is a relaxing color for human mind. Our program has buttons and they change color when you hover over them.





In order to work with mySQL database we used HTML, PHP and CSS to style the website. We ordered the page design with HTML, gave command to the code with PHP and used CSS to style header fonts, colors, background etc.. We used modal box style in some pages to make the page clearer and easier to understand.



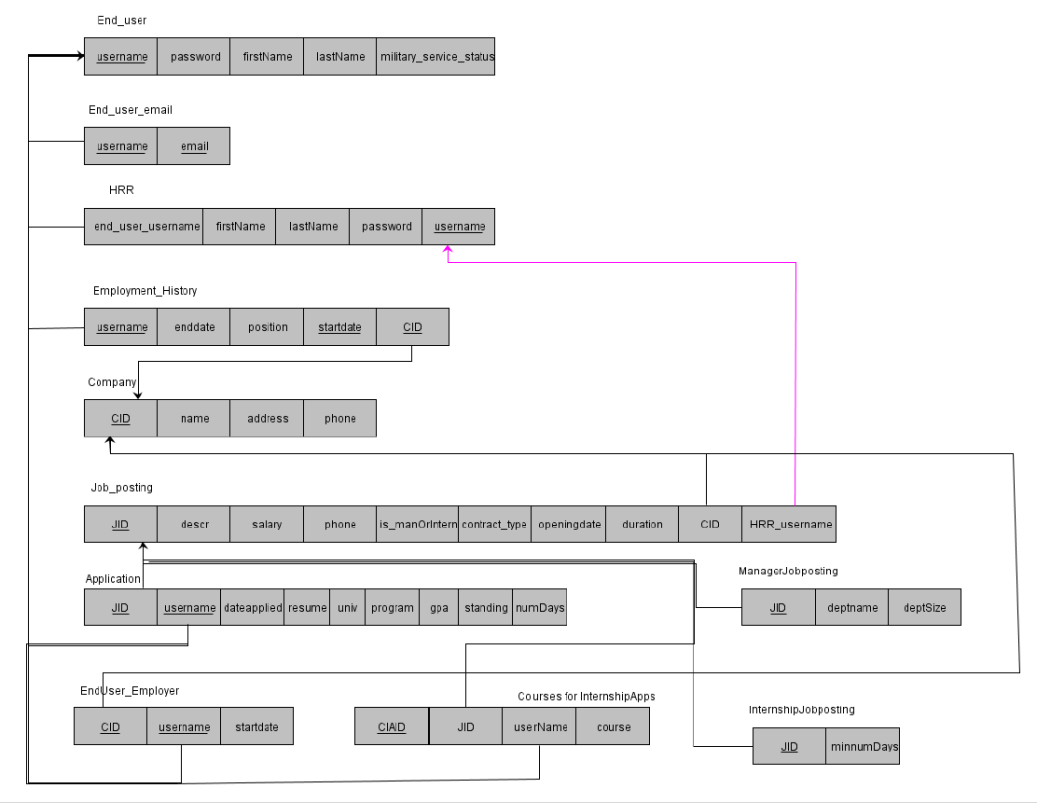
After login page, when you click on the buttons, new pop-up opens and shows what is written on the button. For example:

×

jid	username	fname	lname
4	johnDoe	john	doe
5	johnDoe	john	doe

2- Is the relational schema of your database in 2NF/3NF/BCNF?

In order to answer this question, we observed our relational schema:



First, we looked if this schema supports first normal form. We checked if all attributes depend on the key or not. We checked it there are any composite, multivalued attribute, nested relations and non-atomic tuples. In the description, end-users could have multiple email but first normal form disallows multivalued attribute so we created a new table for end-users' emails as both end-user username and their email as primary key.

For second normal form, we checked if all attributes depend on the whole key. All of our keys in this schema are functionally dependent to their primary key(s).

For third normal form, we checked if all attributes depend on nothing but the key. In our database, we had job postings and some job posting types. If a job posting is a manager job posting, jid would define job type and job type would define department name which is not a candidate key. Since this ISA hierarchy is not allowed in relational schema and the relationship didn't create transitive functional dependency we created new tables for internship job postings and manager job postings.

Lastly, we checked Boyce-Codd normal form. A relation is in BCNF if every determinant is a candidate key. If we look at manager job posting (after 3NF), we can see that jid defines department name which can be a candidate key in this new table.

In conclusion, this database supports all normal forms to BCNF. If it didn't support at least to 3NF, we could have problems with updating, inserting variables and controlling the database.

We knew normal forms were important from the start and did our work according to that knowledge.

3- How would you improve this database?

There are several ways to improve a database:

- **Improve indexing strategies and create optimal indexes**: Indexes are data structures that enable quick selection and sorting of rows in a database table. They speed up data retrieval by allowing for random lookups and easy access to orderly records. Indexing saves our and system's time and effort.
- **Use a stronger CPU**: Better CPU makes the database more efficient and faster.
- **Data defragmentation**: When a large number of records are written to a database over time, the records become fragmented in MySQL's internal data files and on the disk itself. The disk defragmentation will allow the relevant data to be grouped together, allowing I/O-related operations to run faster, which will have a direct impact on overall query and database performance. It is also critical to have enough disk space in general when running a database.
- **Disk type**: Depending on the amount of data the query needs to access for processing and the amount of data returned from the query, fetching the results of even a single query can require millions of I/O operations from disk. As a result, the type of disks in your server can have a significant impact on the performance of your SQL queries. Working with SSD disks can significantly improve overall database performance, particularly SQL query performance.
- **Database version:** Maintaining the most recent version of our database can have a significant impact on overall database performance. One query may perform better in older versions of MySQL than in newer versions, but in terms of overall performance, newer versions usually perform better.
- **Know who has access:** Taking record of the applications and services that have access to our database can be a help in the identification of performance bottlenecks. A single service's poor performance may be slowing down our entire work.