REPORT 2:
GOsling

Team:
Zhunussova Meruyert 200103534
Bekmagambetova Diana 200103329

Task Division
To ensure a smooth and efficient development process, we divided the tasks among ourselves based on our areas of expertise.

Diana is responsible for:

1. Connect to MySQL from our web application.

Connecting to a MySQL database from a web application involves establishing a connection between the web server and the database server through the use of appropriate programming language and database driver. This connection allows for the execution of database operations such as retrieving, updating or deleting data from the database.

2. Create a standalone models package, so that our database logic is reusable and decoupled from our web application.

To ensure that the database logic is reusable and decoupled from the web application, we created a standalone models package. This package contains the database schema and the necessary queries and functions to perform database operations. By decoupling the database logic from the web application, any changes made to the web application will not affect the database logic and vice versa. Additionally, the standalone models package can be used across multiple web applications, reducing the need for duplicative code and improving maintainability of the codebase.

Meruyert is responsible for:

1. Use the appropriate functions in Go's database/sql package to execute different types of SQL statements, and how to avoid common errors that can lead to our server running out of resources.

When executing SQL statements, it is important to be aware of common errors that can lead to our server running out of resources. One of the most common issues is failing to properly close database connections or cursors after executing SQL statements. This can lead to a buildup of unused resources and eventually cause our server to crash.

2. Prevent SQL injection attacks by correctly using placeholder parameters.

Another issue to be aware of is SQL injection attacks. These attacks occur when malicious users are able to insert their own SQL commands into a statement that is executed on the server. To prevent SQL injection attacks, it is important to correctly use placeholder parameters when executing SQL statements. Placeholder parameters can help sanitize user input and prevent the injection of malicious SQL commands.

Future work:

We’re going to concentrate on displaying the dynamic data from our MySQL database in some proper HTML pages.

Conclusion
Our progress on the GOsling project can be tracked on our GitHub repository at https://github.com/mistledi/GOsling.