

## REPORT 2: GOsling

### Team:

Zhunussova Meruyert 200103534

Bekmagambetova Diana 200103329

### Diana:

1. End-To-End Testing (end-to-end testing should give us more confidence that our application is working correctly than unit testing in isolation.)  
I adapt our TestPing function so that it runs an end-to-end test on our code. Specifically, I want the test to ensure that a GET /ping request to our application calls the ping handler function and results in a 200 OK status code and "OK" response body.
2. Mocking Dependencies  
Created a new pkg/models/mock package and two new snippets.go and users.go files to hold the database model mocks.  
changed our application struct in cmd/web/main.go so that it uses interfaces instead, which are satisfied by both my mock and production database models.  
wrote some tests for our showSnippet handler and GET /snippet/:id route.  
For the request GET /snippet/1 I received a 200 OK response with the relevant mocked snippet in the HTML response body. For all other requests to GET /snippet/\* I've received a 404 Not Found response.

### Meruyert:

1. Add an end-to-end test for the POST /user/signup route, which is handled by our signup User handler.
2. Make a GET /user/signup request. This will return a response which contains a CSRF cookie in the response headers and a CSRF token in the HTML response body.
3. Extract the CSRF token from the HTML response body.
4. Make a POST /user/signup request, using the same http. Client that we used in step 2 and including the CSRF token alongside the other POST data that we want to test.
5. Integration test to ensure that our mysql.UserModel.Get() method is working correctly. To improve confidence in our application even more and our production MySQL database models will work as expected.

Progress can be tracked on our GitHub repository at <https://github.com/mistledi/GOsling>.