Redwanul Mutee (rm4243)

Farhad Ahmed (fa961)

**Application Security Lab 2: Vulnerable Application**

**GitHub link:** <https://github.com/redwan97/ApplicationSecurity/tree/master/Lab2>

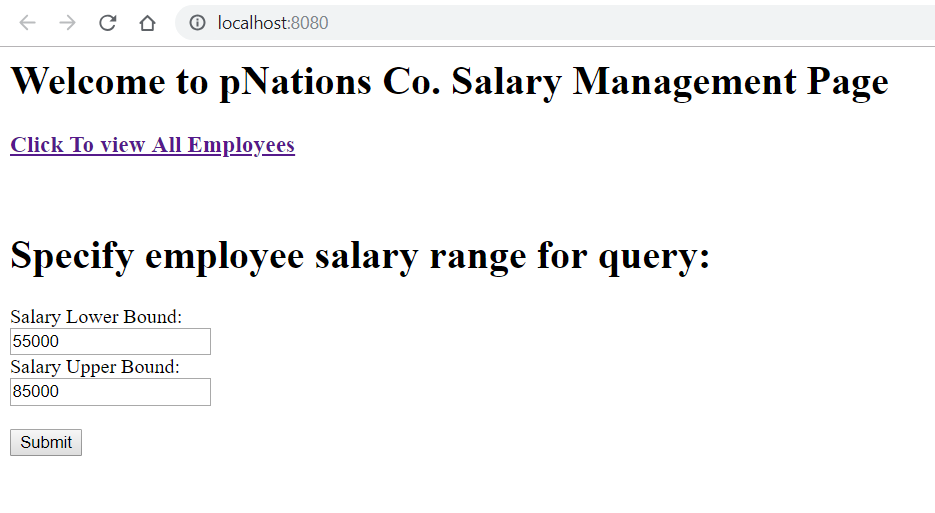
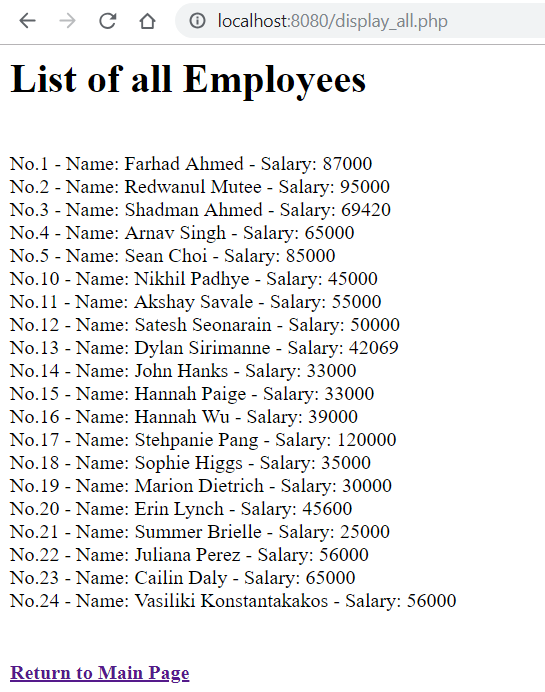
**Instructions:**

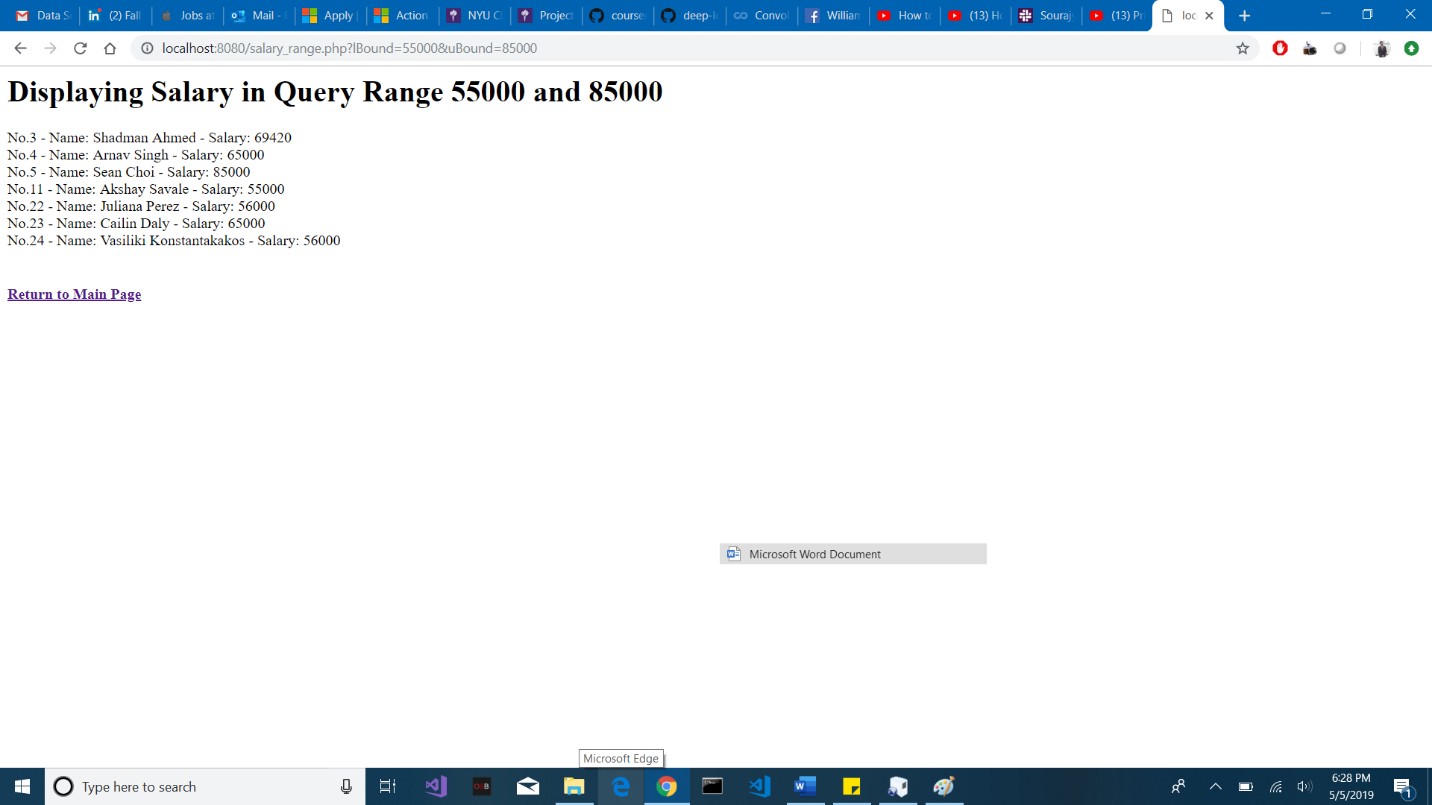
(1) run usbwebserver.exe on an windows machine

(2) click localhost in window that opens up

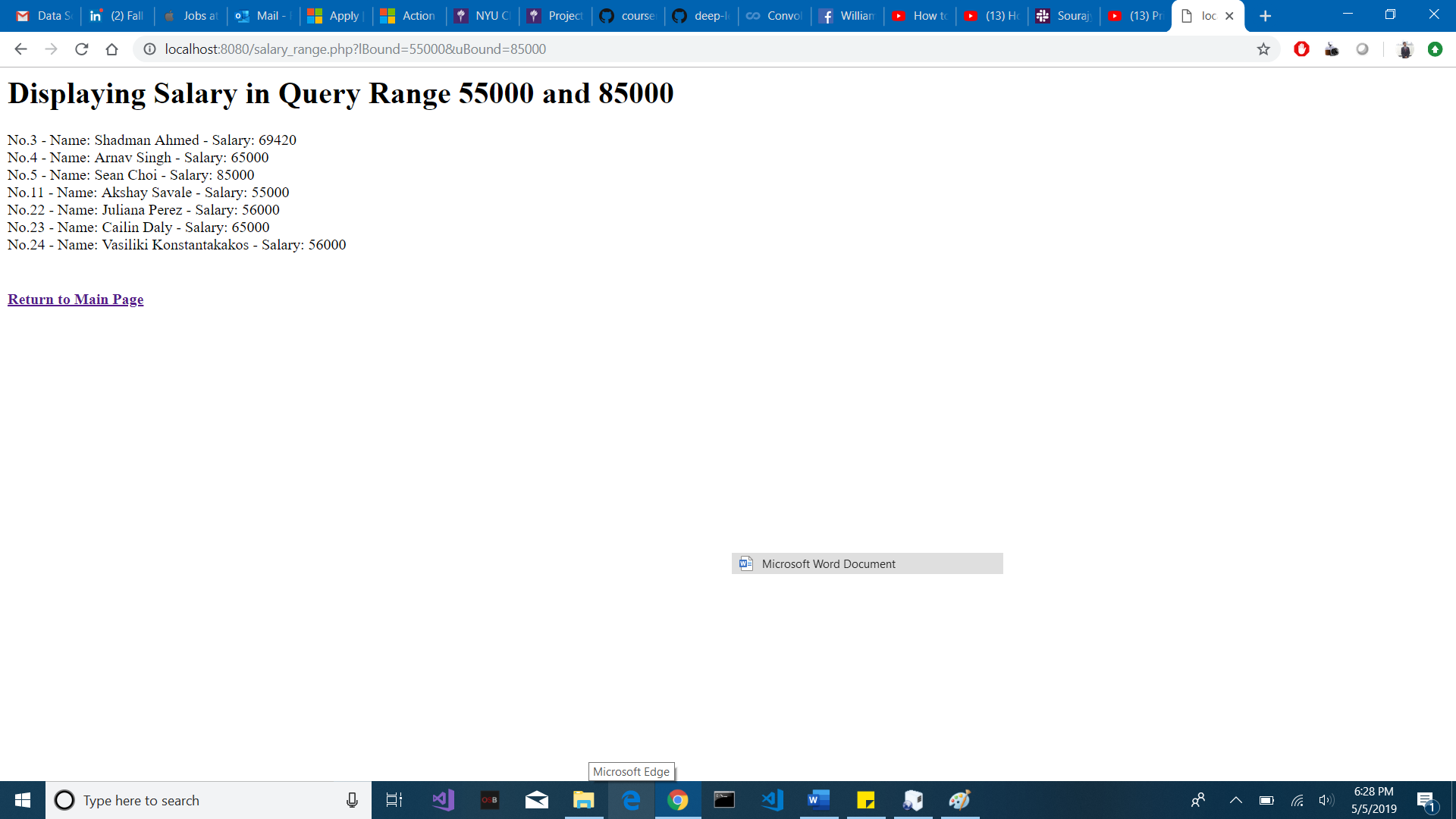
(3) explore vulnerabilities detailed below

For Lab 2, we were instructed to create an application with some vulnerabilities. To make it easy to have vulnerabilities we decided to use a WAMP (windows, Apache, MySql, PHP) server without any security. The source of the WAMP server is usbwebserver. Our application is supposed to be a salary management service that allows a supervisor to view the salaries of current employees and conduct a simple query on a database of employees. When the submit button is pressed, the query results are shown on the next page.





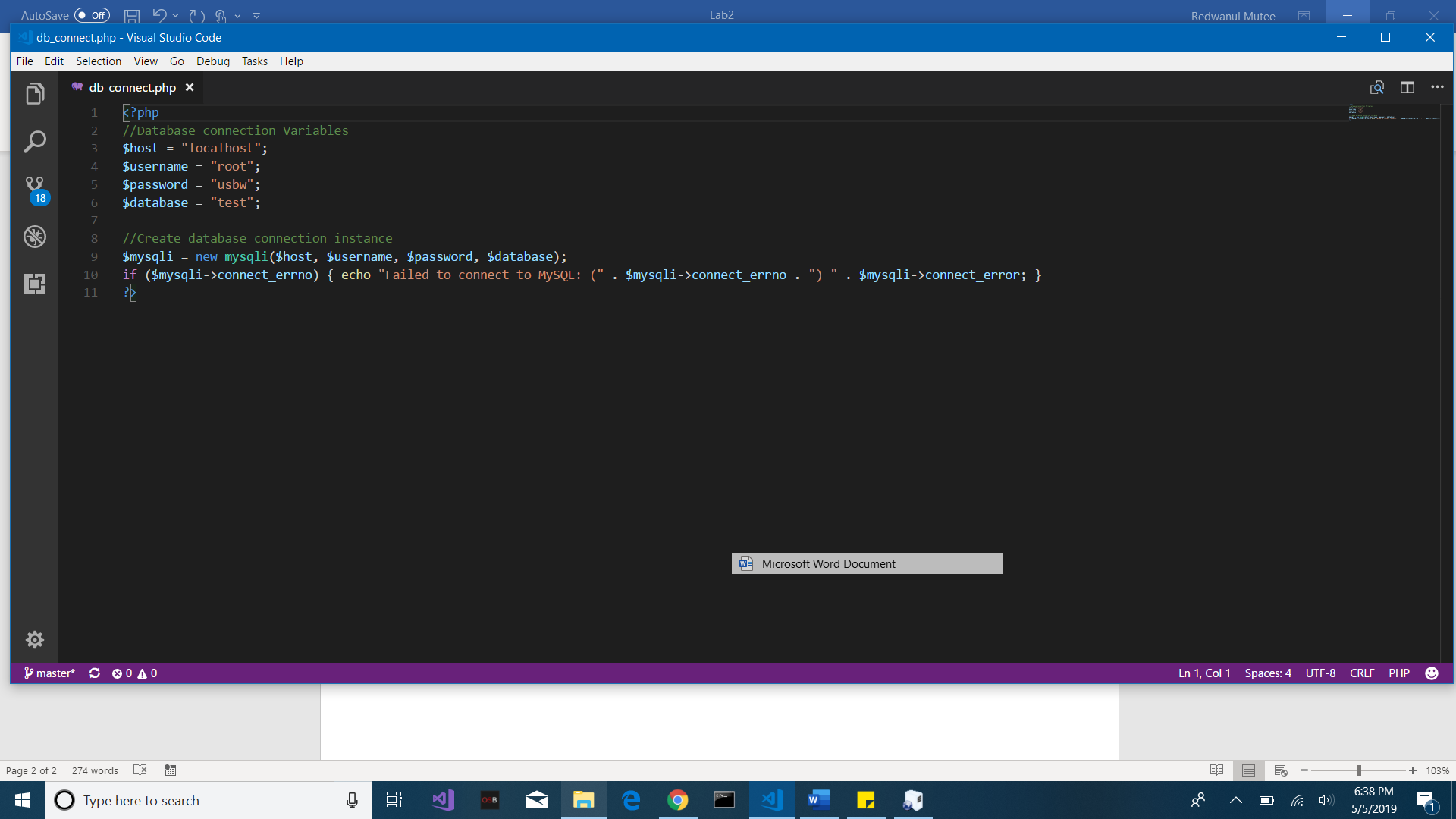
Because no form of security was placed in the application, it is relatively simple to carry out CSRF or SQL-injection attacks. A malicious user could use SQL-injections to potentially edit the database or just drop the entire database among many other malicious actions. This attack can be carried out by creating a creative SQL expression and entering it into the form instead of specifying the salary bounds. A malicious user could also carry out CSRF attacks by analyzing the urls generated on submit of the form and editing them to access otherwise inaccessible information.

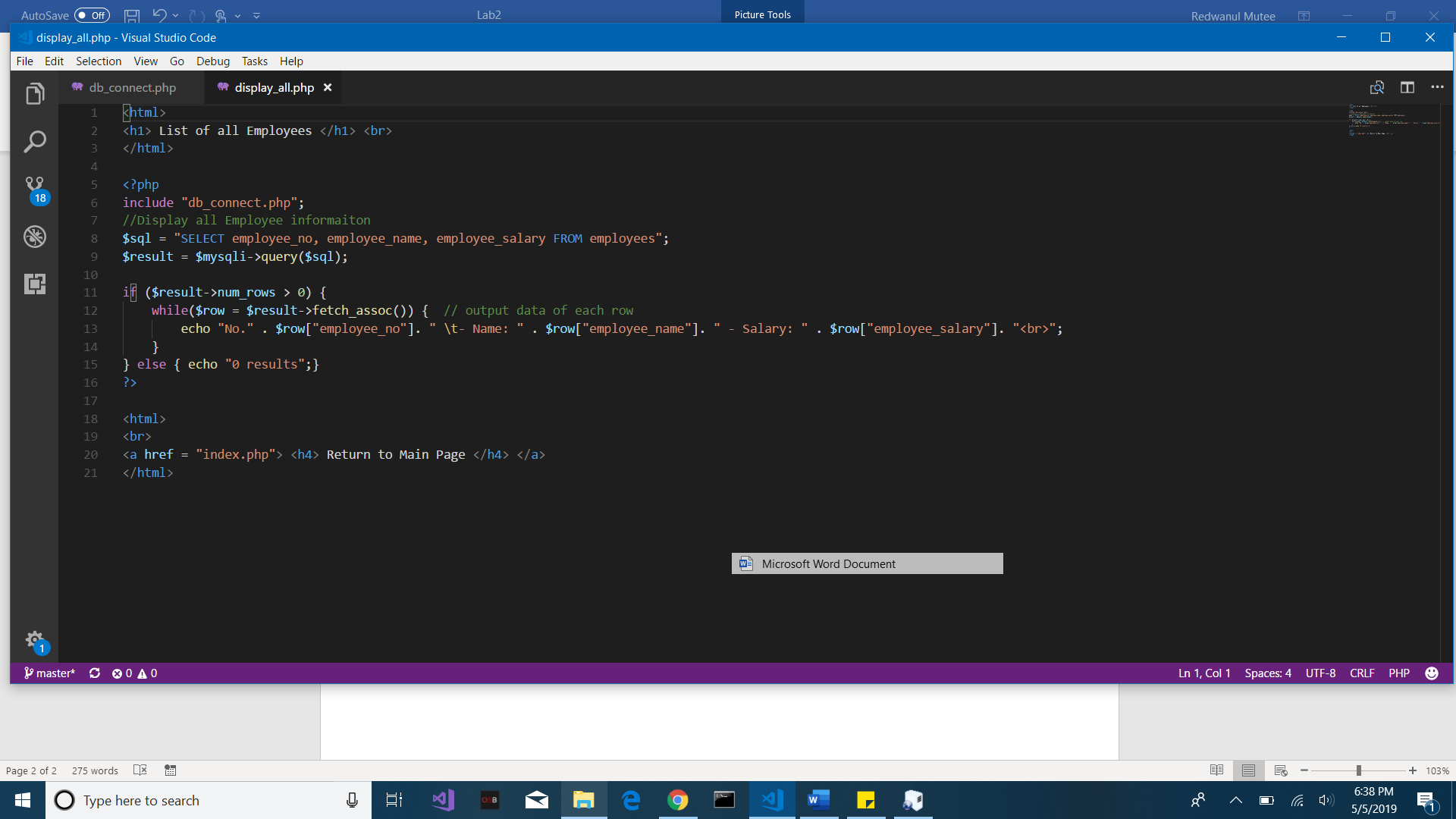


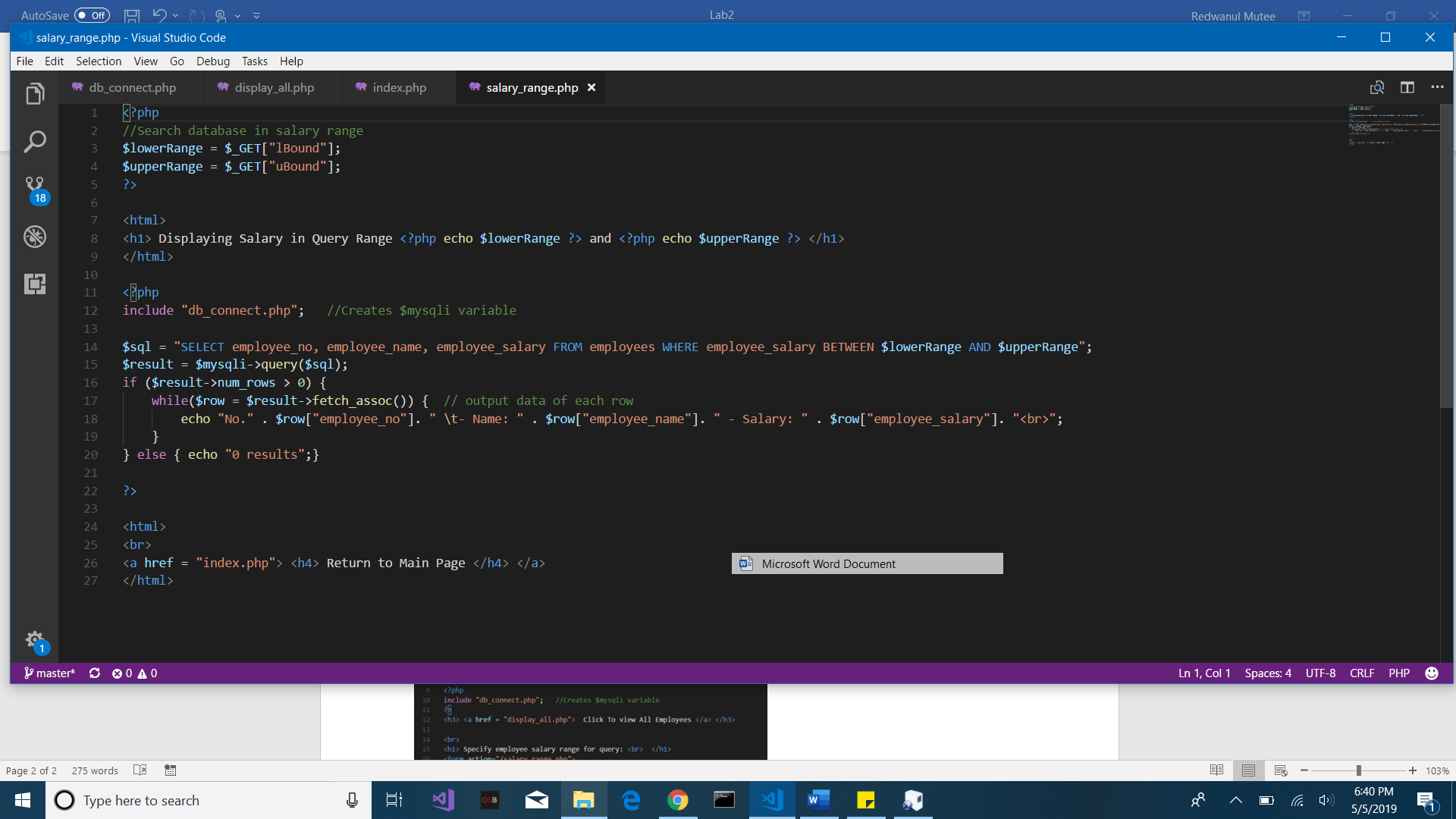
In the future we plan to switch from a WAMP server to the Django framework as it makes it easier to implement security. Django allows for CSRF tokens to be displayed very easily and directly into html code. Django provides an abstraction over the chosen database and makes it very difficult to carry out injection

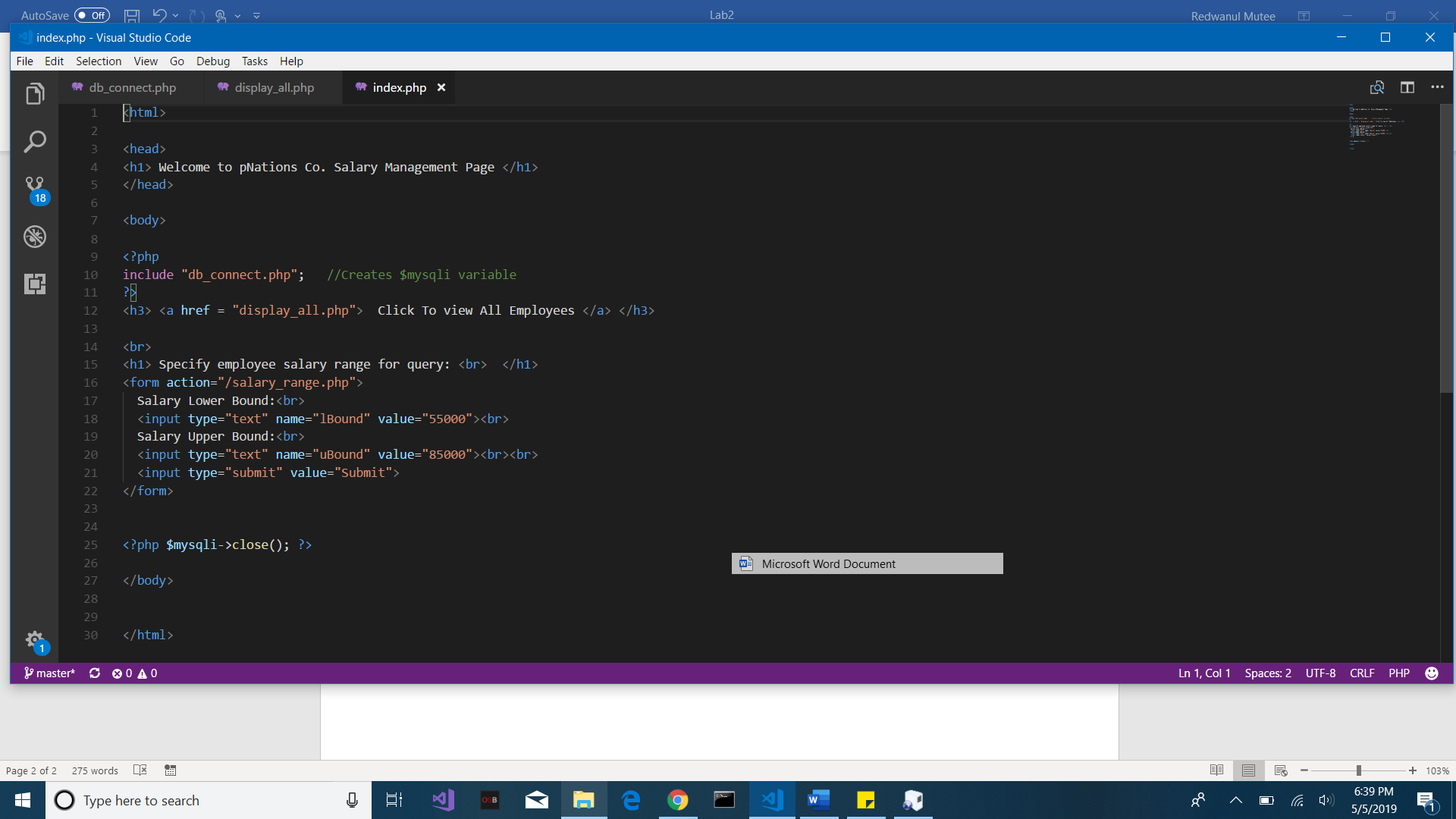
attacks.

**CODE:**







s