

Experiment No. 9

Title: Network Sniffing - Wireshark

**Batch: B-2 Roll No.: 16010422234 Experiment No.: 9**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Aim:** To perform network sniffing using wire shark tool

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Questions:**

1. **What is the difference between Burp suite and Wire shark tools?**

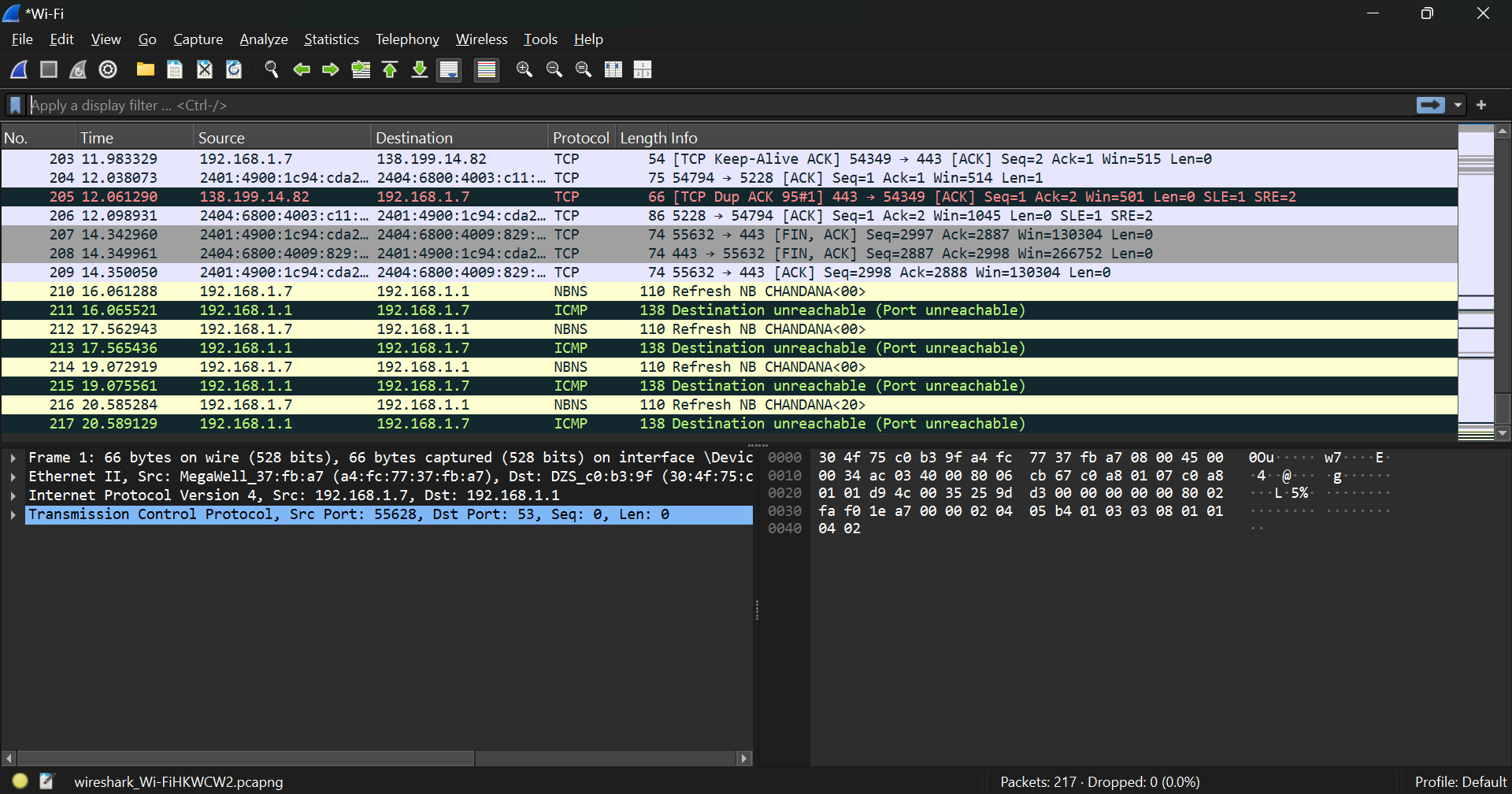
* Wireshark is primarily a network protocol analyzer used for network troubleshooting, analysis, software and protocol development, and education. It captures and displays packets in real-time and can interpret the data flowing in and out of a network.
* Burp Suite, on the other hand, is more focused on web application security testing. It functions as an intercepting proxy, allowing a user to see and modify the traffic between a browser and the server. It is used for security testing of web applications and can manipulate web traffic to test the security parameters of the application.

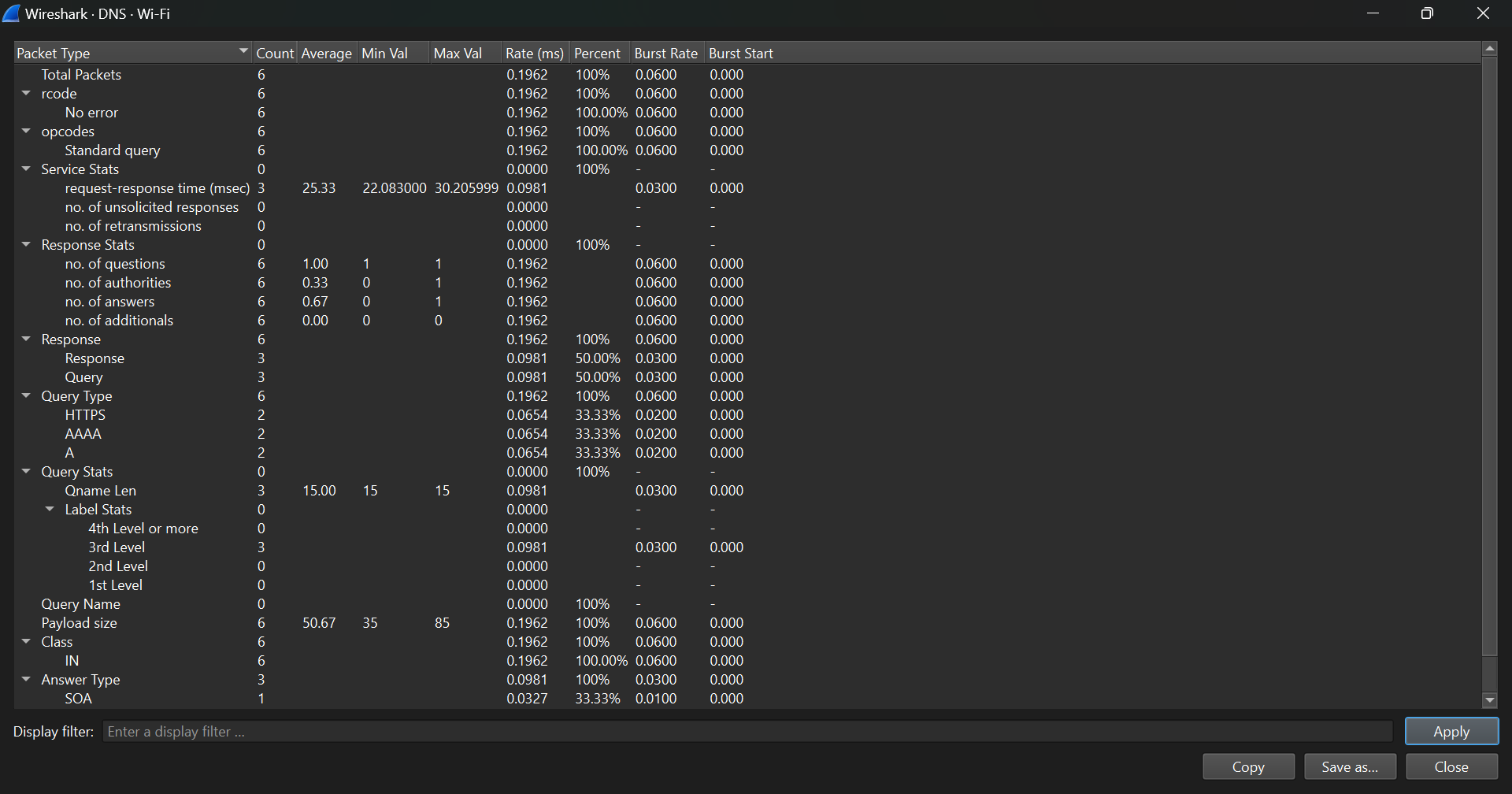
1. **Suggest the methods and/or security mechanisms to protect the password being leaked using tools like wireshark.**

* Use Encryption: Employ TLS/SSL for all sensitive communications. Encrypted traffic cannot be easily read by packet sniffers if intercepted.
* Secure Authentication Protocols: Implement secure authentication mechanisms such as OAuth, which do not transmit passwords directly.
* HTTPS: Ensure that all login pages and data transmissions use HTTPS to secure web traffic.
* Strong Password Policies: Enforce complex passwords that are difficult to decode even if packet data is captured.
* Regular Updates and Patches: Keep all network software and devices updated to protect against vulnerabilities that might be exploited to capture network traffic.



**Result:**

****

****

**CREATE DATABASE IF NOT EXISTS UserDB;**

**USE UserDB;**

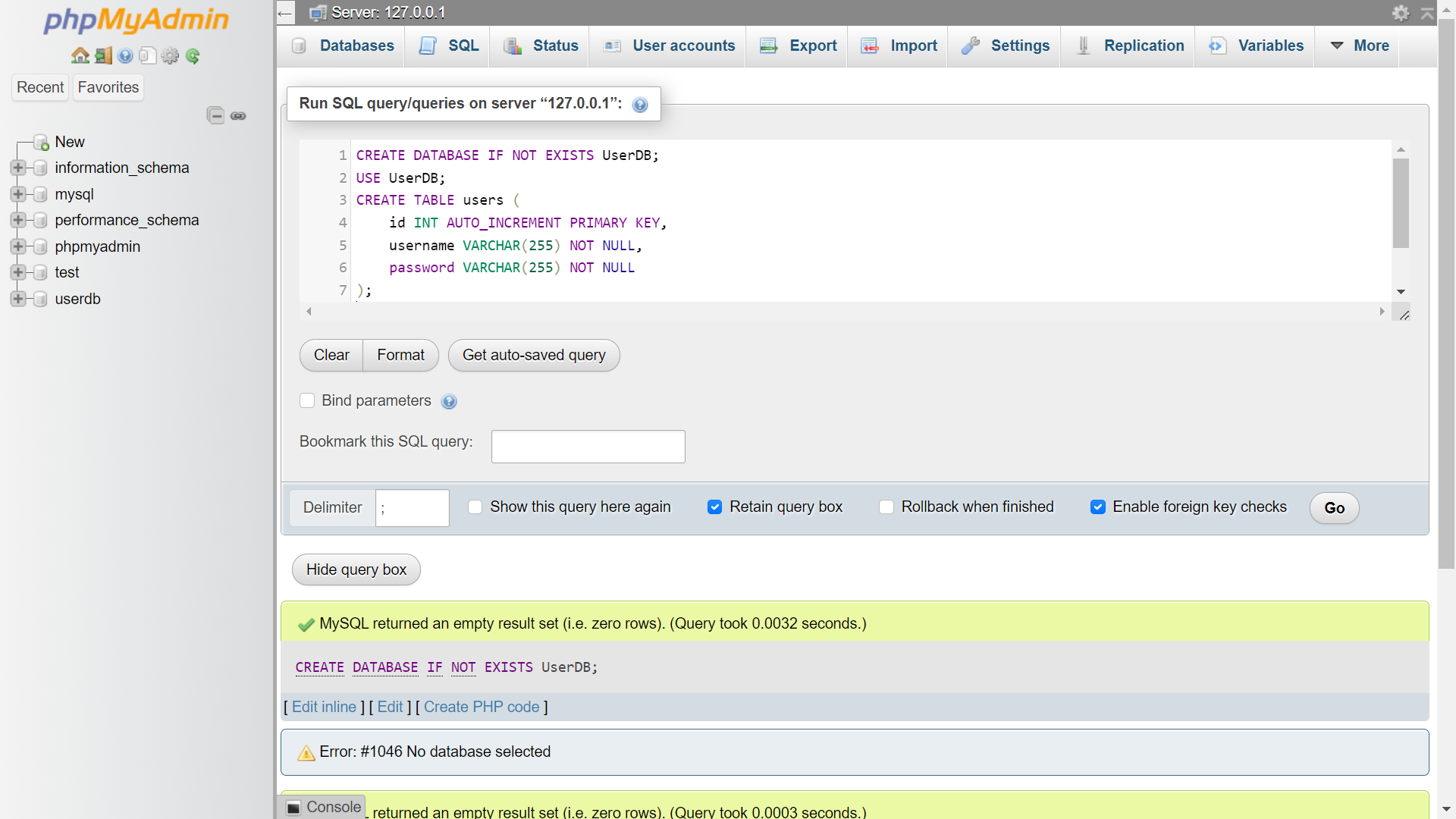
**CREATE TABLE users (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**username VARCHAR(255) NOT NULL,**

**password VARCHAR(255) NOT NULL**

**);**

****

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Register</title>**

**</head>**

**<body>**

**<h2>Registration Form</h2>**

**<form action="register.php" method="get">**

**Username: <input type="text" name="username"><br>**

**Password: <input type="password" name="password"><br>**

**<input type="submit" value="Register">**

**</form>**

**</body>**

**</html>**

**<?php**

**$servername = "localhost";**

**$username = "root"; // Update your database username**

**$password = ""; // Update your database password**

**$dbname = "UserDB";**

**$conn = new mysqli($servername, $username, $password, $dbname);**

**if ($conn->connect\_error) {**

**die("Connection failed: " . $conn->connect\_error);**

**}**

**// Insecure: Password should be hashed in a real application**

**$sql = "INSERT INTO users (username, password)**

**VALUES ('".$\_GET['username']."', '".$\_GET['password']."')";**

**if ($conn->query($sql) === TRUE) {**

**echo "New record created successfully";**

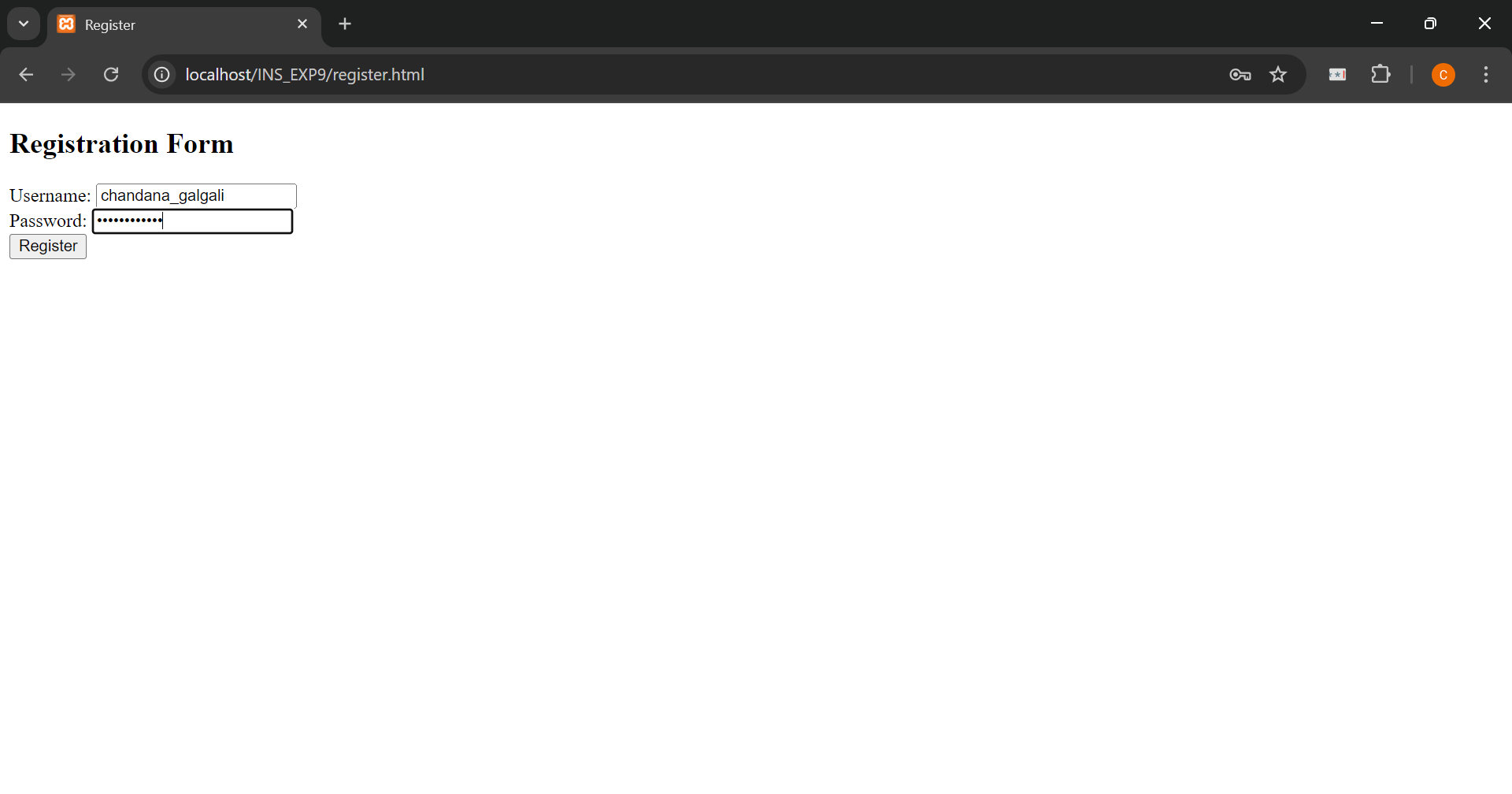
**} else {**

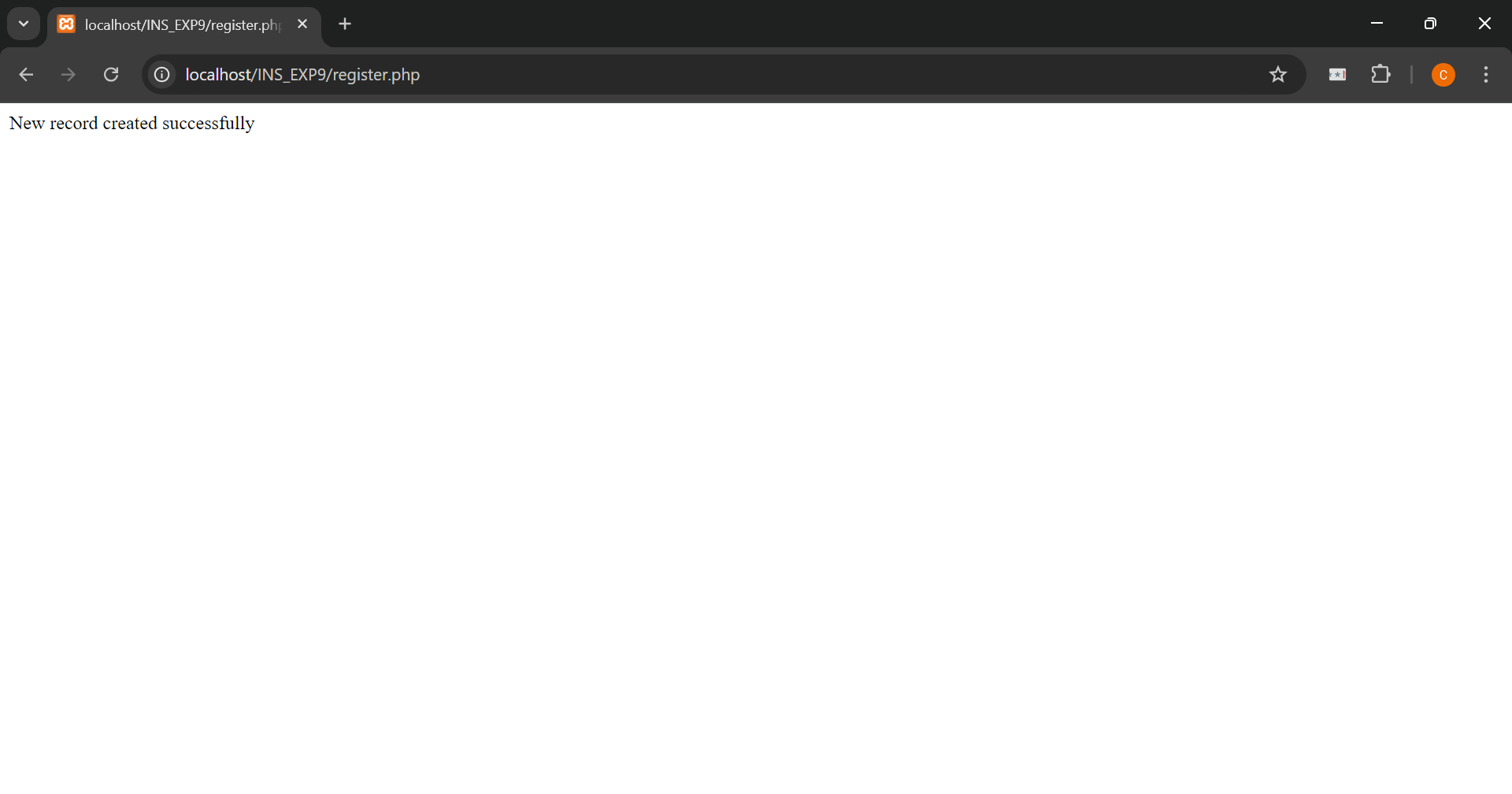
**echo "Error: " . $sql . "<br>" . $conn->error;**

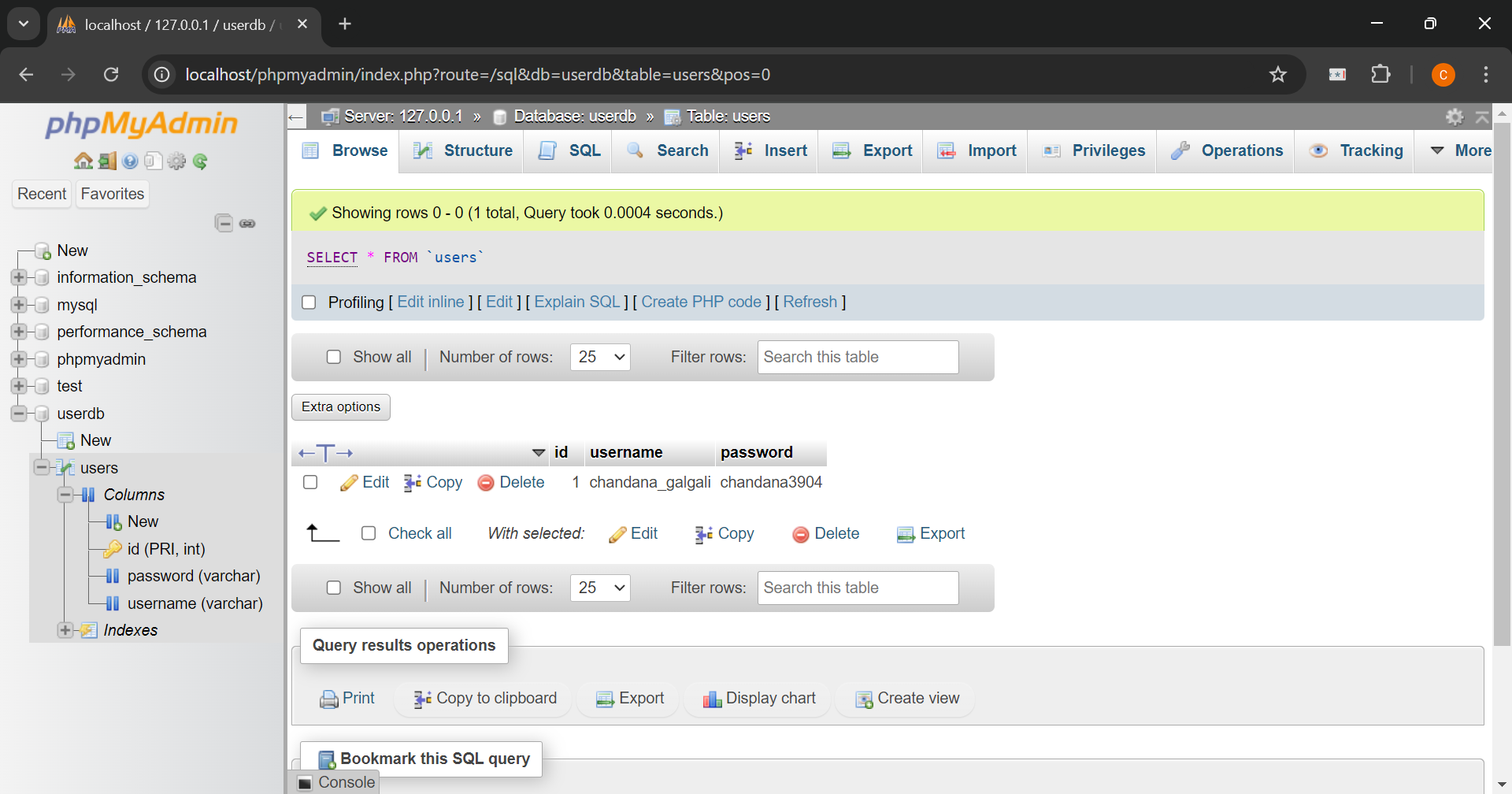
**}**

**$conn->close();**

**?>**

****

****

****

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Login</title>**

**</head>**

**<body>**

**<h2>Login Form</h2>**

**<form action="login.php" method="get">**

**Username: <input type="text" name="username"><br>**

**Password: <input type="password" name="password"><br>**

**<input type="submit" value="Login">**

**</form>**

**</body>**

**</html>**

**<?php**

**$servername = "localhost";**

**$username = "root";**

**$password = "";**

**$dbname = "UserDB";**

**$conn = new mysqli($servername, $username, $password, $dbname);**

**if ($conn->connect\_error) {**

**die("Connection failed: " . $conn->connect\_error);**

**}**

**$sql = "SELECT id FROM users WHERE username = '".$\_GET['username']."' AND password = '".$\_GET['password']."'";**

**$result = $conn->query($sql);**

**if ($result->num\_rows > 0) {**

**echo "Login successful";**

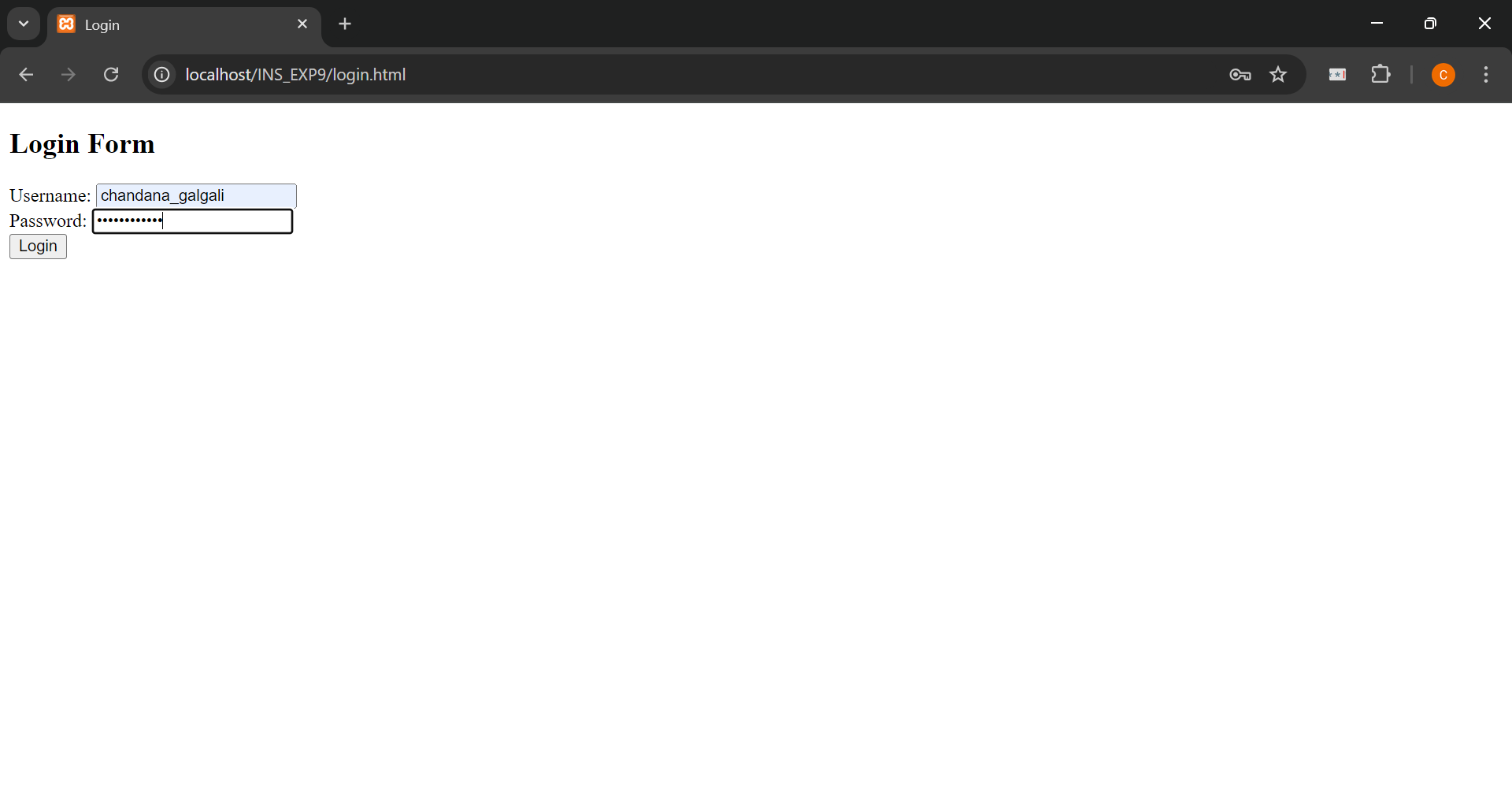
**} else {**

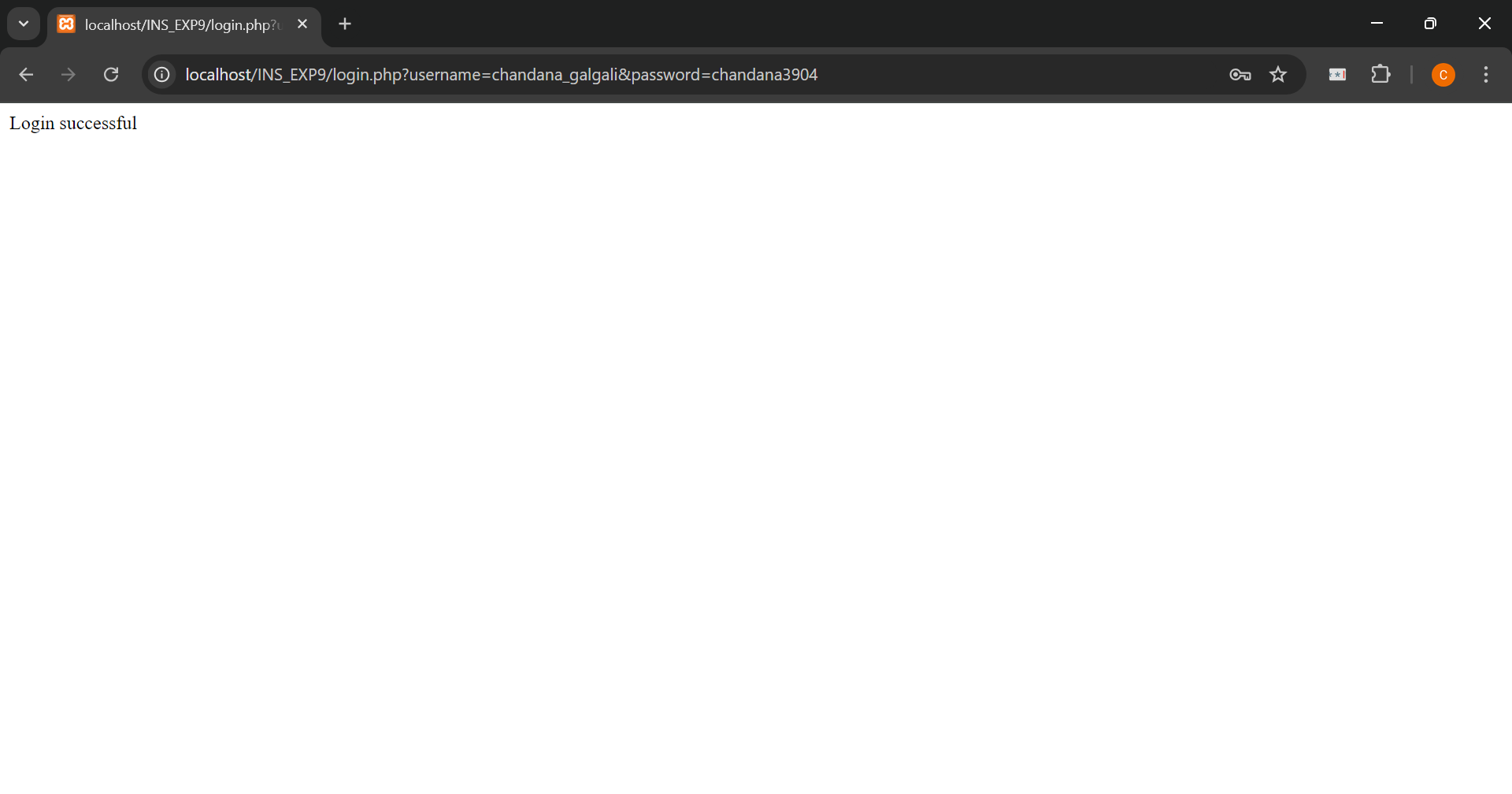
**echo "Invalid username or password";**

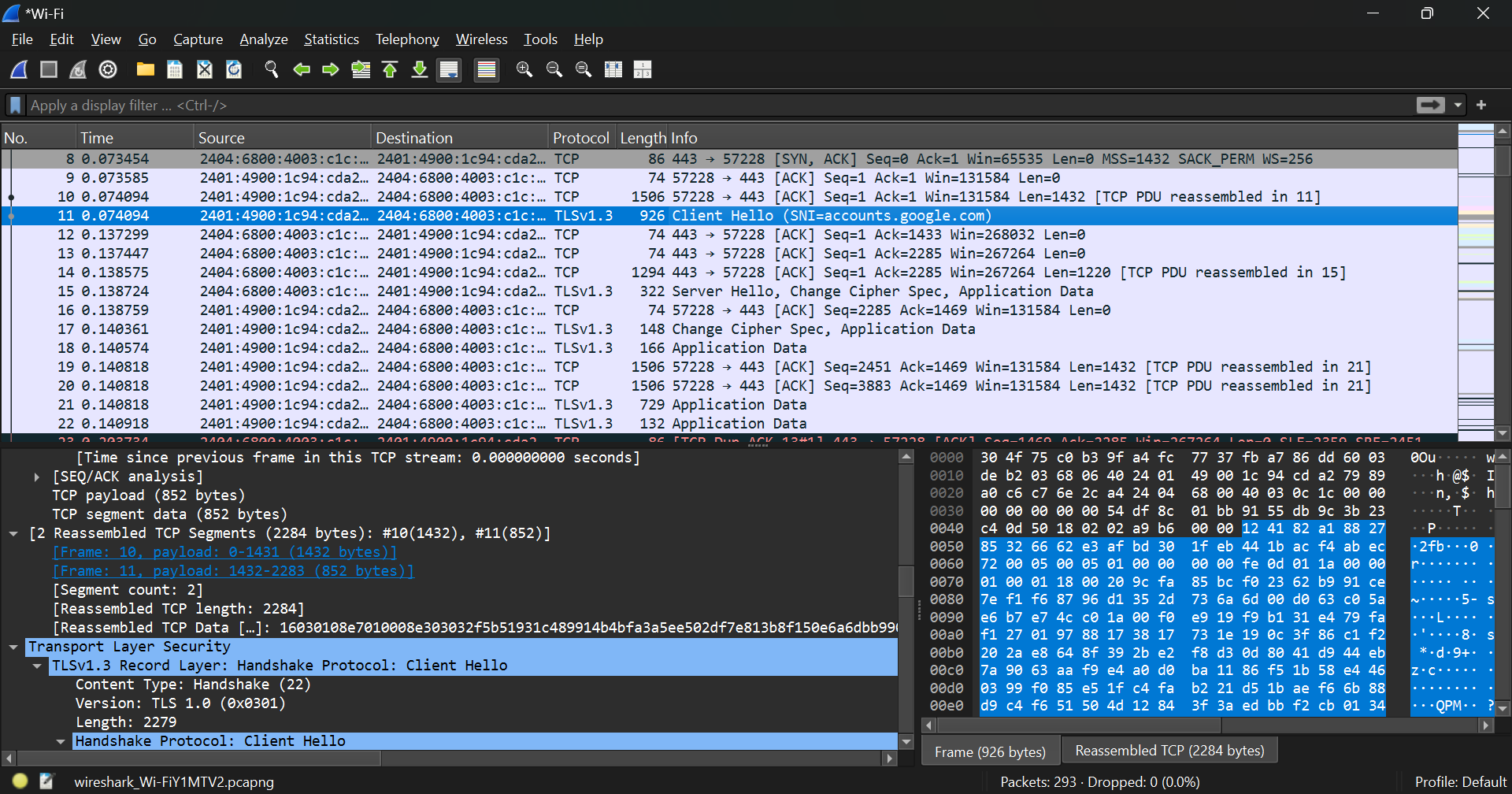
**}**

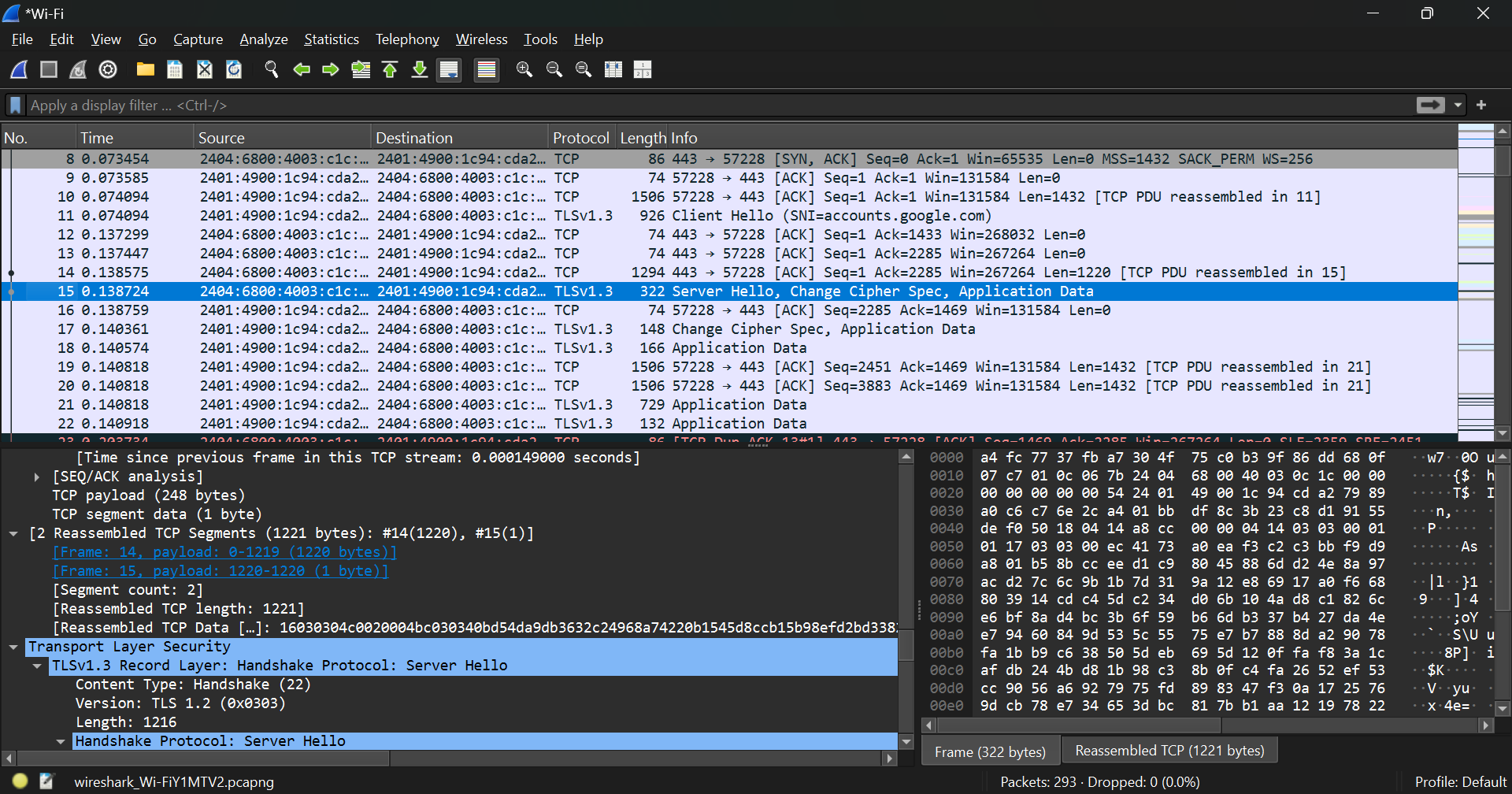
**$conn->close();**

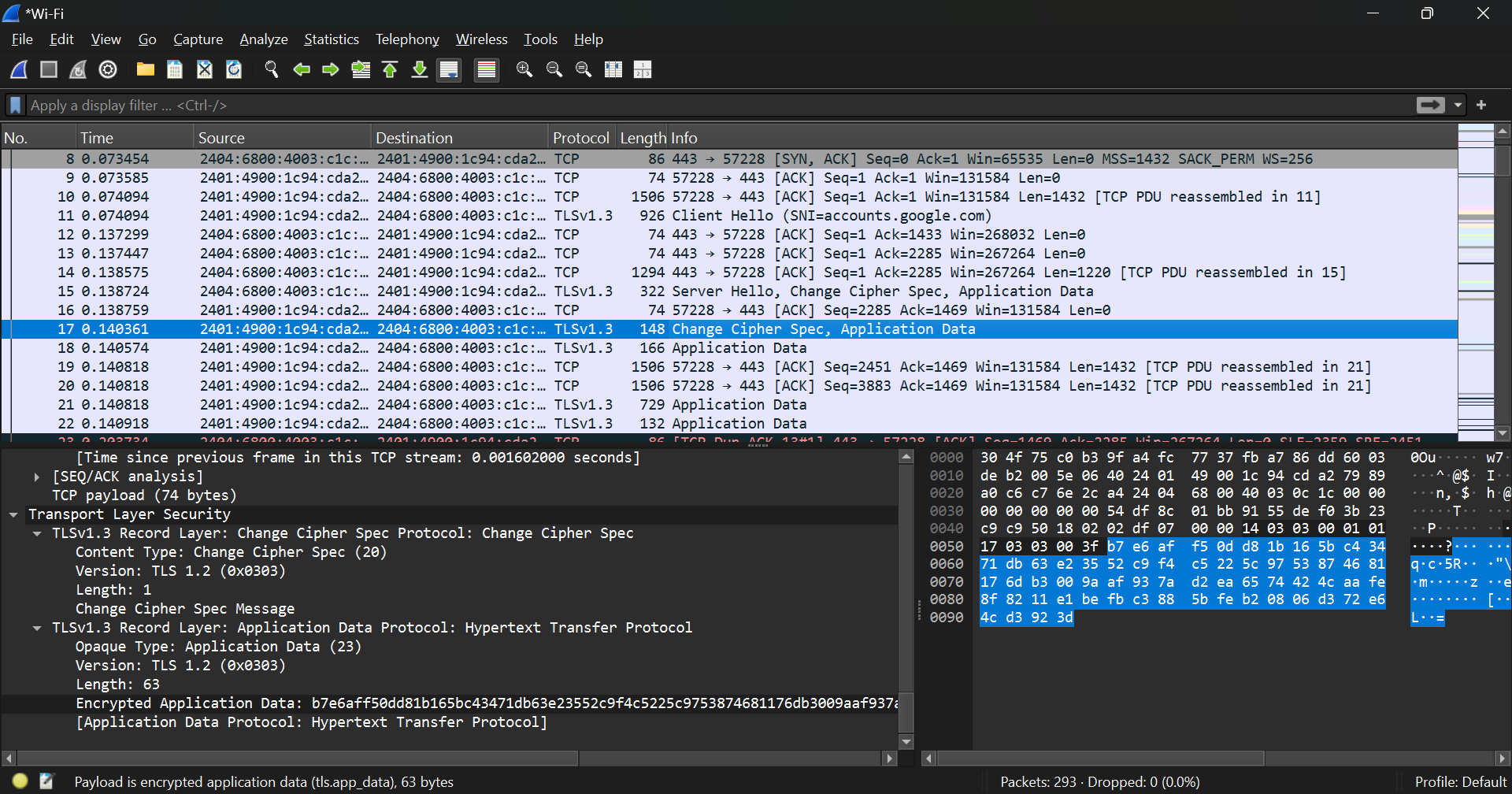
**?>**

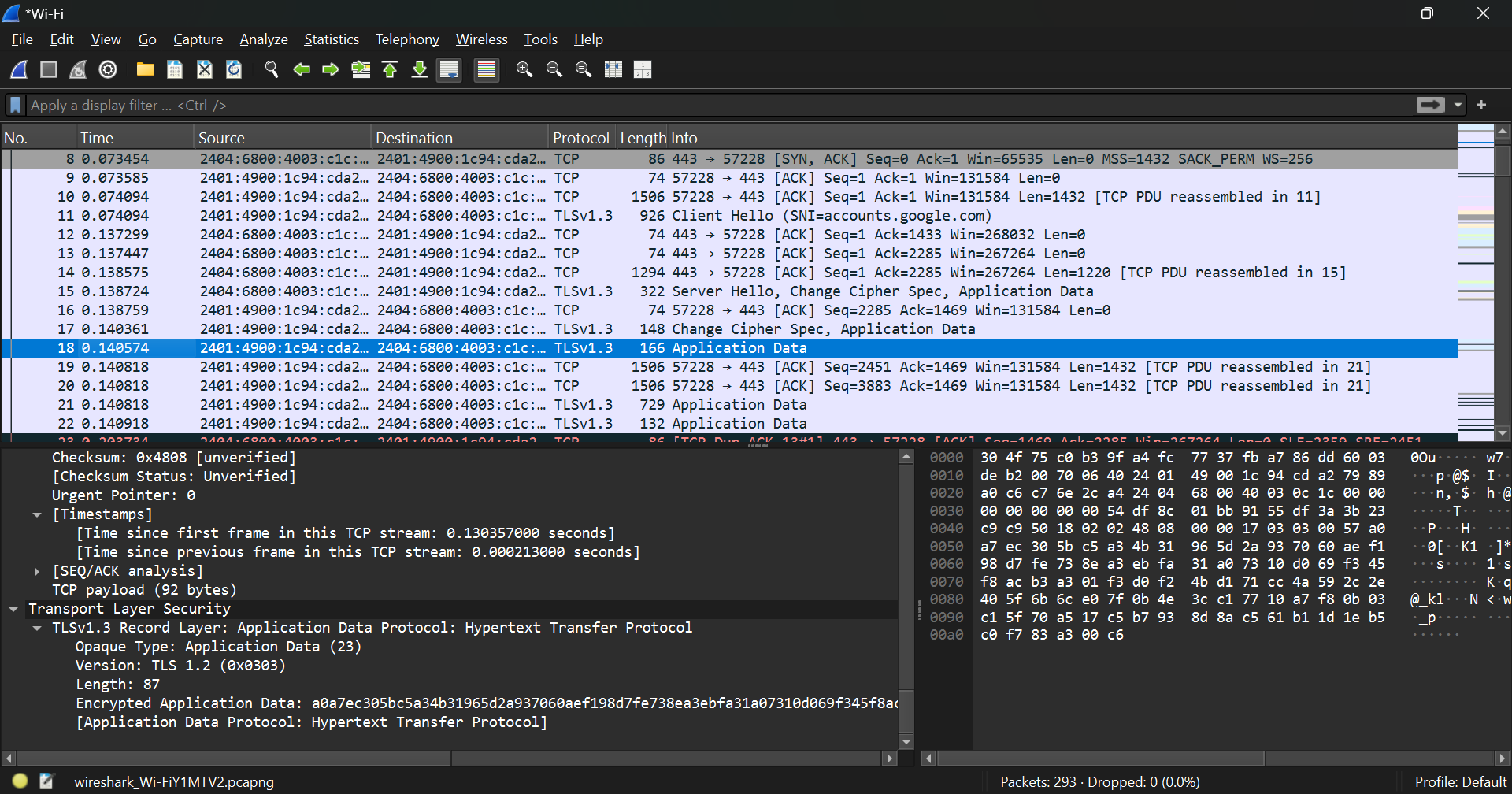
****

****

****

****

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:** Understand Security issues related to Software, Web and Networks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

The Network Sniffing experiment using Wireshark provided profound insights into the workings of network packets and their flow across various interfaces. By observing the transmission of data in real-time, students gain a hands-on understanding of how protocols interact and how data moves through a network. This experiment underlines the importance of network security measures and the role of sophisticated tools like Wireshark in diagnosing, troubleshooting, and ensuring the integrity of network data. The skills acquired from this practical application are crucial for future network administrators and cybersecurity professionals in safeguarding against potential network vulnerabilities and attacks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**