Assignment5

Team: 12

Team Members: AJ Kimbrough ANK210005, Ariana Qozat AOQ210000, Aryan Tyagi AXT200084, Khushboo Amarnani KAA210004, Natthiya Sae Ngow NXS220110.

(a)

```
<!-- Login Form -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
</head>
<body>
    <h2>Login</h2>
    <form action="getdata.php" method="GET">
        <label for="EID">Username:</label>
        <input type="text" id="EID" name="EID" value="admin" required><br><br>

        <label for="Password">Password:</label>
        <input type="password" id="Password" name="Password" value="password" required><br><br>

        <button type="submit">Login</button>
    </form>
</body>
</html>
```

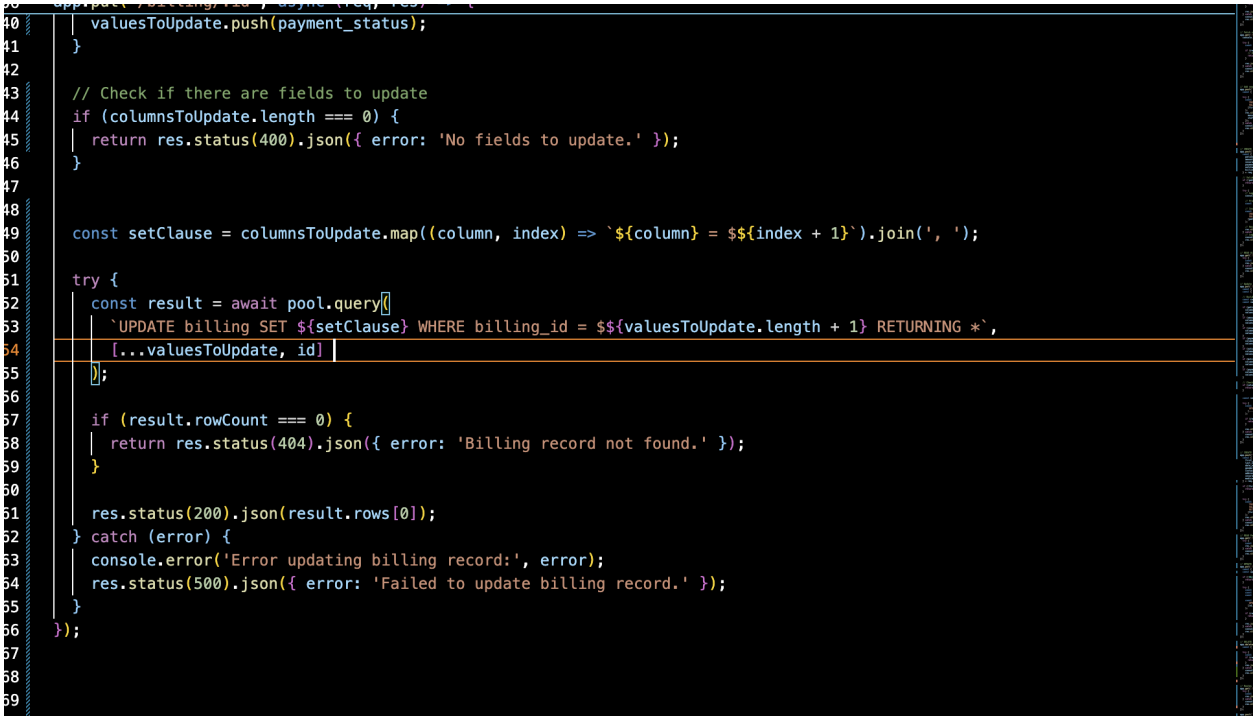
(b) **Login**

```
// Login route with SQL injection vulnerability

app.post('/login', async (req, res) => {
  const { username, password } = req.body;

  const query = `SELECT * FROM users WHERE username = '${username}' AND password = '${password}'`;

  try {
    const result = await pool.query(query);

    if (result.rows.length > 0) {
      res.json({ message: 'Login successful!' });
    } else {
      res.status(401).json({ error: 'Invalid credentials' });
    }
  } catch (err) {
    console.error('Error with login:', err);
    res.status(500).json({ error: 'Error logging in' });
  }
});
```

Update Command (Billing Component)

```
// Update billing information by ID (Injection)
app.put('/billing/:id', async (req, res) => {
  const { id } = req.params;
  const { patient_id, amount_due, insurance_coverage, payment_received, outstanding_balance, billing_date, payment_status } =

  // Build an array of columns to update
  const columnsToUpdate = [];
  const valuesToUpdate = [];

  if (patient_id) {
    columnsToUpdate.push('patient_id');
    valuesToUpdate.push(patient_id);
  }
  if (amount_due) {
    columnsToUpdate.push('amount_due');
    valuesToUpdate.push(amount_due);
  }
  if (insurance_coverage) {
    columnsToUpdate.push('insurance_coverage');
    valuesToUpdate.push(insurance_coverage);
  }
  if (payment_received) {
    columnsToUpdate.push('payment_received');
    valuesToUpdate.push(payment_received);
  }
  if (outstanding_balance) {
    columnsToUpdate.push('outstanding_balance');
    valuesToUpdate.push(outstanding_balance);
  }
  if (billing_date) {
```

```
    columnsToUpdate.push('payment_received');
    valuesToUpdate.push(payment_received);
  }
  if (outstanding_balance) {
    columnsToUpdate.push('outstanding_balance');
    valuesToUpdate.push(outstanding_balance);
  }
  if (billing_date) {
    columnsToUpdate.push('billing_date');
    valuesToUpdate.push(billing_date);
  }
  if (payment_status) {
    columnsToUpdate.push('payment_status');
    valuesToUpdate.push(payment_status);
  }

  // Check if there are fields to update
  if (columnsToUpdate.length === 0) {
    return res.status(400).json({ error: 'No fields to update.' });
  }


  const setClause = columnsToUpdate.map((column, index) => `${column} = $${index + 1}`).join(', ');

  try {
    const result = await pool.query(
      `UPDATE billing SET ${setClause} WHERE billing_id = $${valuesToUpdate.length + 1} RETURNING *`,
      [...valuesToUpdate, id]
    );
```



```
    valuesToUpdate.push(payment_status);
  }

  // Check if there are fields to update
  if (columnsToUpdate.length === 0) {
    return res.status(400).json({ error: 'No fields to update.' });
  }


  const setClause = columnsToUpdate.map((column, index) => `${column} = $${index + 1}`).join(', ');

  try {
    const result = await pool.query(
      `UPDATE billing SET ${setClause} WHERE billing_id = $${valuesToUpdate.length + 1} RETURNING *`,
      [...valuesToUpdate, id]
    );

    if (result.rowCount === 0) {
      return res.status(404).json({ error: 'Billing record not found.' });
    }

    res.status(200).json(result.rows[0]);
  } catch (error) {
    console.error('Error updating billing record:', error);
    res.status(500).json({ error: 'Failed to update billing record.' });
  }
});
```

(C)

```php
<!-- Prepared Statement -->

<?php
$eid = $_GET['EID'];
$pwd = $_GET['Password'];

$conn = new mysqli("localhost", "root", "password", "medical_database");

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT Name, Role, Department FROM staff WHERE eid = ? AND password = ?";

if ($stmt = $conn->prepare($sql)) {
    $stmt->bind_param("ss", $eid, $pwd);
    $stmt->execute();
    $stmt->bind_result($name, $role, $department);

    while ($stmt->fetch()) {
        printf("Name: %s -- Role: %s -- Department: %s\n", $name, $role, $department);
    }

    $stmt->close();
}

$conn->close();
?>
```