IST 615 – Cloud Management

Scalable and Secure Sports Club E-Commerce Platform

Team members

Heet Gala – hgala02@syr.edu

Abhishek Shinde – ashinde@syr.edu

Divyaraj Sarvaiya – dvsarvai@syr.edu

Project Objectives:

The primary objective of this project is to develop a safe, scalable, and high-performing web platform for a sports club to sell sports-related merchandise. The platform aims to provide a user-friendly and secure online shopping experience while leveraging cloud services to ensure optimal performance and scalability.

The key objectives are as follows:

- Create a visually appealing and responsive website for showcasing and selling sports merchandise. This includes designing an intuitive user interface, implementing responsive design techniques, and ensuring cross-browser compatibility.

- Implement robust security measures to protect user information and financial transactions. This involves implementing industry-standard security protocols such as HTTPS, data encryption, secure authentication mechanisms, and regular security audits.

- Utilize cloud services to achieve high availability, scalability, and efficient resource management. By leveraging cloud infrastructure, the platform can easily scale resources up or down based on demand, ensuring optimal performance during peak traffic periods, and minimizing costs during low traffic periods.

- In our project, Azure Data Studio was used to design and manage the database schema, create tables for storing product and user data, and define relationships between different entities. Complex queries required for retrieving and manipulating data were developed and optimized using Azure Data Studio's query execution and performance tuning tools.

- Implement user authentication and authorization mechanisms for secure access and account management. This includes features such as user login, password reset, and role-based access control to ensure that users can only access and perform authorized actions.

Configuration of Cloud Services:

The project utilizes the following Azure cloud services:

Azure Monitor / Log Analytics:

Log Analytics is a component of Azure Monitor that helps collect, analyze, and query log data from various sources, including applications and services.

Metric Data: Metric Data refers to the numerical data collected by Azure Monitor, which can be used for monitoring performance, resource utilization, and other metrics.

Diagnostic logs: These components likely refer to the logs and diagnostic data generated by the application or services, which can be analyzed for troubleshooting, auditing, or monitoring purposes.

Azure SQL Server:

Our team opted to deploy an Azure SQL Database instance using Azure Data Studio to host the relational database for our e-commerce platform. This database serves as the repository for crucial data such as user information, product details, and order data, essential for the platform's operations. Leveraging Azure SQL Database affords us automated backups, automatic failover mechanisms, and scalability features, ensuring robust data durability and high availability.

Azure Data Studio:

Moreover, Azure Data Studio empowers us to configure the database with precision, selecting appropriate performance tiers and service levels tailored to our platform's requirements. Additionally, we utilize advanced features such as data encryption at rest and in transit, along with auditing and threat detection functionalities, to bolster data security and integrity further.

Azure Web App:

Azure Web App Service is a fully managed platform that enables developers to build, deploy, and scale web applications quickly and efficiently. It supports multiple programming languages, including .NET, Java, Node.js, Python, and PHP, allowing developers to use their preferred language and framework.

With Azure Web App Service, developers can focus on writing code and building features without worrying about managing infrastructure. The platform handles tasks such as provisioning servers, deploying code, and load balancing, streamlining the development and deployment process.

Tasks Completed:

Front-end Development: We've built and designed the login and registration pages of our website using HTML, CSS, and JavaScript. These pages serve as the entry points for users to access our platform.

Thorough Testing: To ensure our login and registration pages work seamlessly, we've extensively tested them. We hosted them on Azure Virtual Machine and containers within the Azure environment. This testing process involved checking if users can input their information correctly, ensuring that the pages look good and function well on various devices like phones, tablets, and computers, and making sure that everything works smoothly, from submitting forms to handling any errors that may arise.

Client-side Form Validation: We have implemented techniques to validate the information users input into our forms. This ensures that users provide the necessary information in the correct format before proceeding further.

Responsive Design: Our front-end pages are designed to adapt to different screen sizes and devices. Whether users access our platform from a smartphone, tablet, or desktop computer, the pages adjust accordingly for the best viewing experience.

Branding Consistency: We have styled our front-end pages using CSS to match the sports club's branding and design guidelines. This ensures that our platform maintains a cohesive and visually appealing look that aligns with the sports club's identity, providing users with a consistent and engaging experience.

Issues Encountered:

Integration of Azure SQL Database:

Integrating Azure SQL Database with login and registration functionality required careful planning to ensure data security and integrity. The team encountered challenges in mapping database schemas to application requirements, implementing efficient data storage and retrieval mechanisms, and establishing secure connections between the application and the database. Ensuring compatibility between the database structure and the application's data model was crucial for seamless integration, but complexities arose due to differences in data types, constraints, and indexing strategies.

Deployment Coordination:

Managing the deployment of multiple cloud services across different Azure resources posed challenges, necessitating careful planning and orchestration. The team encountered difficulties in coordinating the deployment of Azure VMs, SQL Database instances, and other cloud services to ensure seamless integration and functionality. Challenges included managing dependencies between services, configuring networking and security settings, and automating deployment processes to ensure consistency and reliability. Lack of experience with deployment orchestration tools and practices added complexity to the process, requiring additional time and effort to streamline deployment workflows and minimize deployment errors.

Lessons Learned:

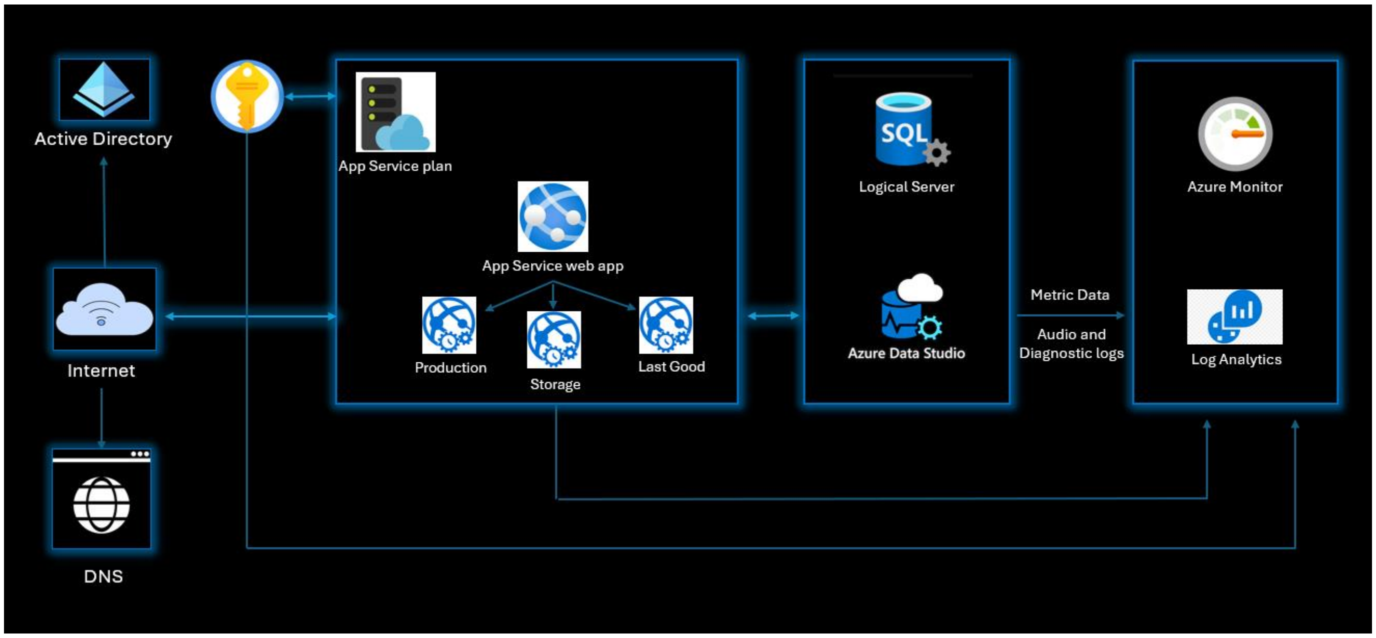
Understanding Cloud Services:

A thorough understanding of cloud services and their configurations is essential. Consulting proper documentation and online resources helps overcome learning curves. As none of us had prior experience with cloud infrastructure, we had to rely heavily on available resources to grasp the concepts and functionalities of Azure services. This taught us the importance of investing time in learning and understanding cloud technologies to effectively leverage them for our project.

Secure Database Integration:

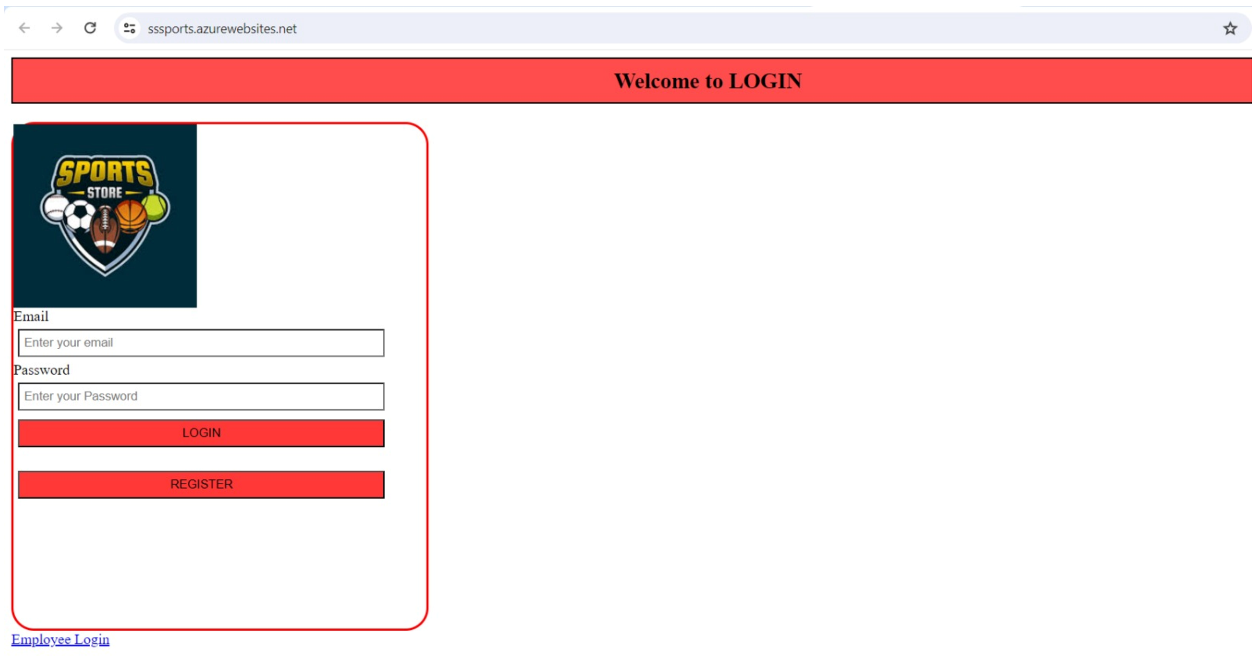
Following best practices for secure database integration, such as proper input validation and encryption, ensures data security and compliance with regulations like GDPR. Integrating Azure SQL Database was a new experience for us, and we learned the importance of implementing robust security measures to protect sensitive data. Through research and trial-and-error, we gained insights into database security best practices and the importance of safeguarding user information.

Architecture Diagram:



U/I Front end:

Login page:


sssports.azurewebsites.net

Welcome to LOGIN

Email

Enter your email

Password

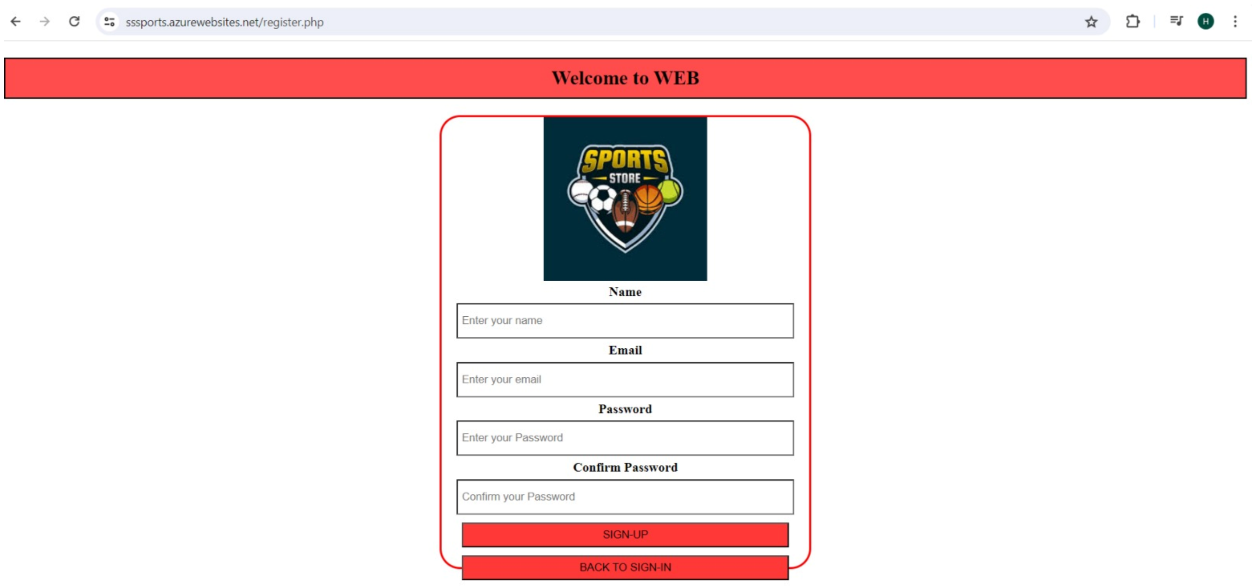
Enter your Password

LOGIN

REGISTER

Employee Login

Register page:


sssports.azurewebsites.net/register.php

Welcome to WEB

Name

Enter your name

Email

Enter your email

Password

Enter your Password

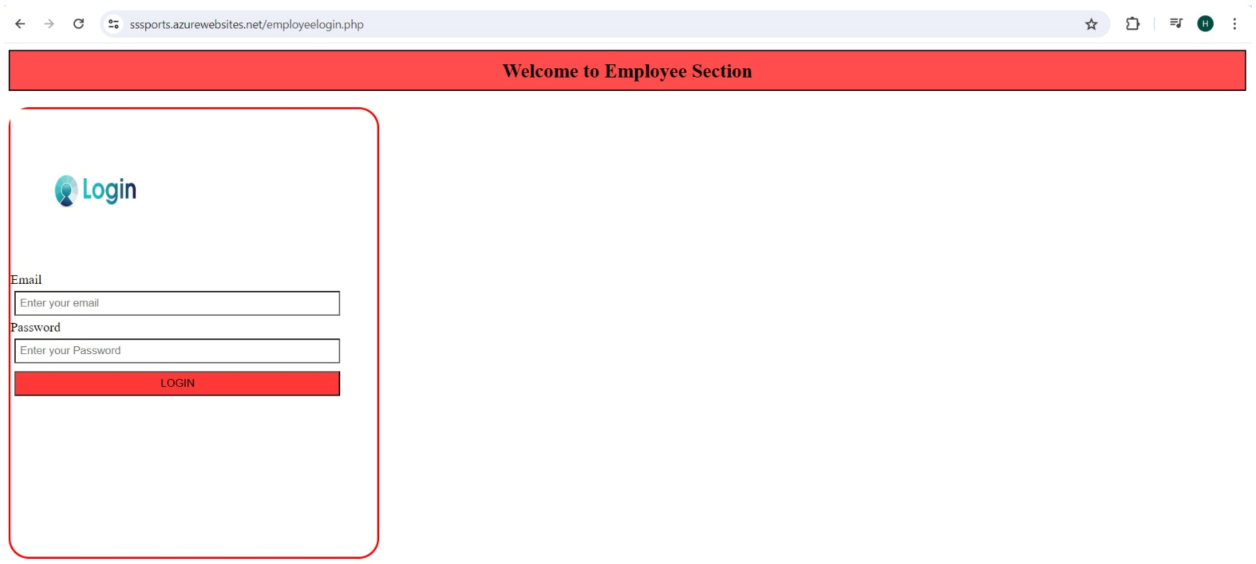
Confirm Password

Confirm your Password

SIGN-UP

BACK TO SIGN-IN

Employee login page:



sssports.azurewebsites.net/employeelogin.php

Welcome to Employee Section

Login

Email

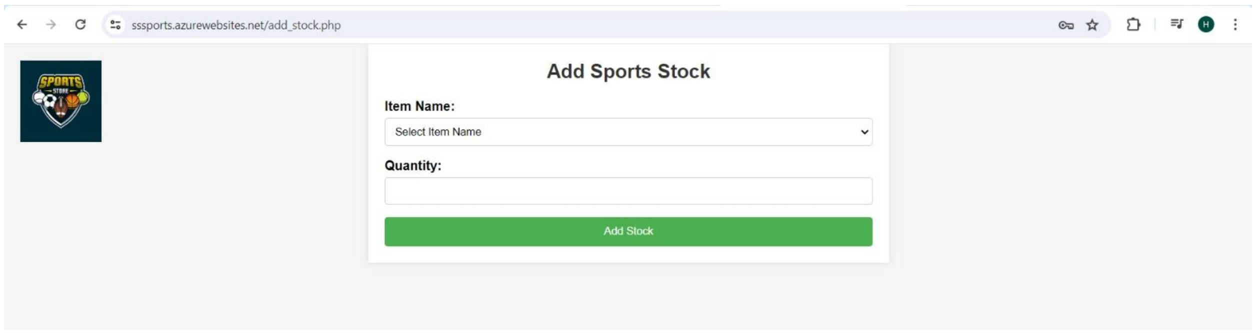
Enter your email

Password

Enter your Password

LOGIN

Add stock / products page:



sssports.azurewebsites.net/add_stock.php

Add Sports Stock

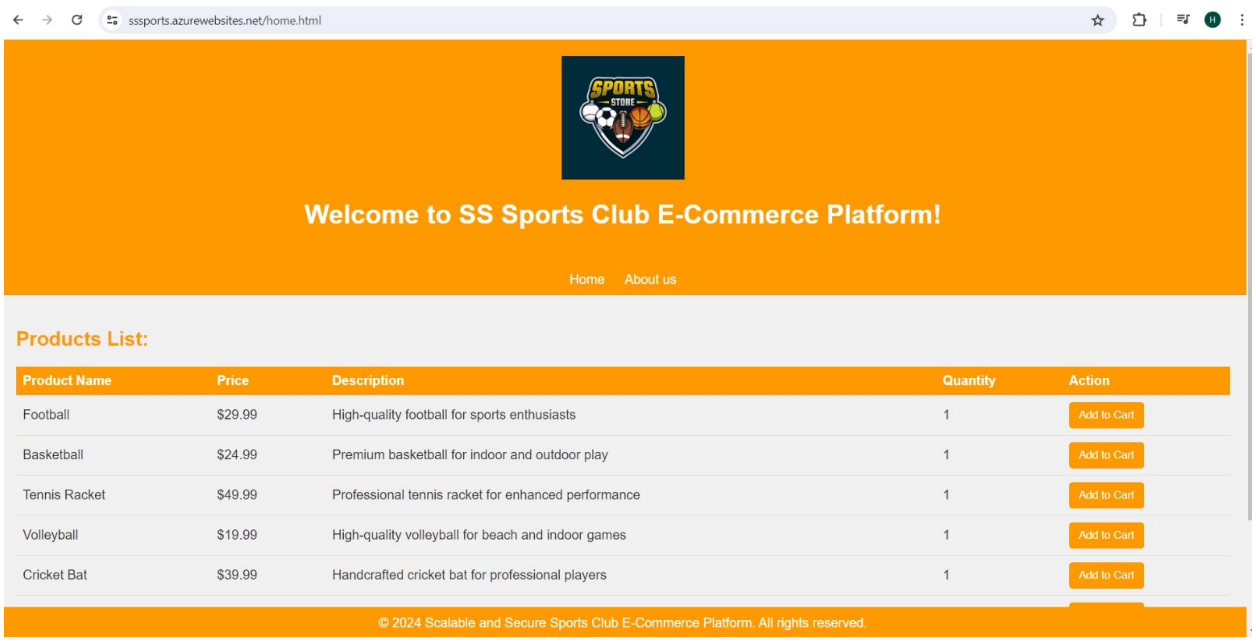
Item Name:

Select Item Name

Quantity:

Add Stock

Home page:



sssports.azurewebsites.net/home.html

Welcome to SS Sports Club E-Commerce Platform!

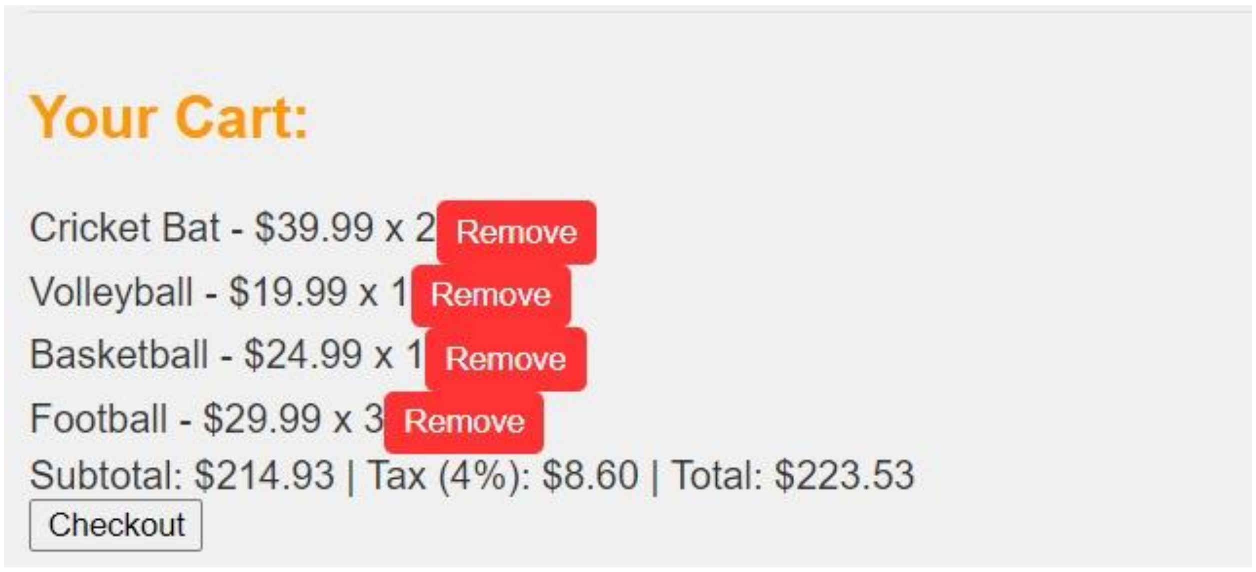
Home About us

Products List:

Product Name	Price	Description	Quantity	Action
Football	$29.99	High-quality football for sports enthusiasts	1	Add to Cart
Basketball	$24.99	Premium basketball for indoor and outdoor play	1	Add to Cart
Tennis Racket	$49.99	Professional tennis racket for enhanced performance	1	Add to Cart
Volleyball	$19.99	High-quality volleyball for beach and indoor games	1	Add to Cart
Cricket Bat	$39.99	Handcrafted cricket bat for professional players	1	Add to Cart

© 2024 Scalable and Secure Sports Club E-Commerce Platform. All rights reserved.

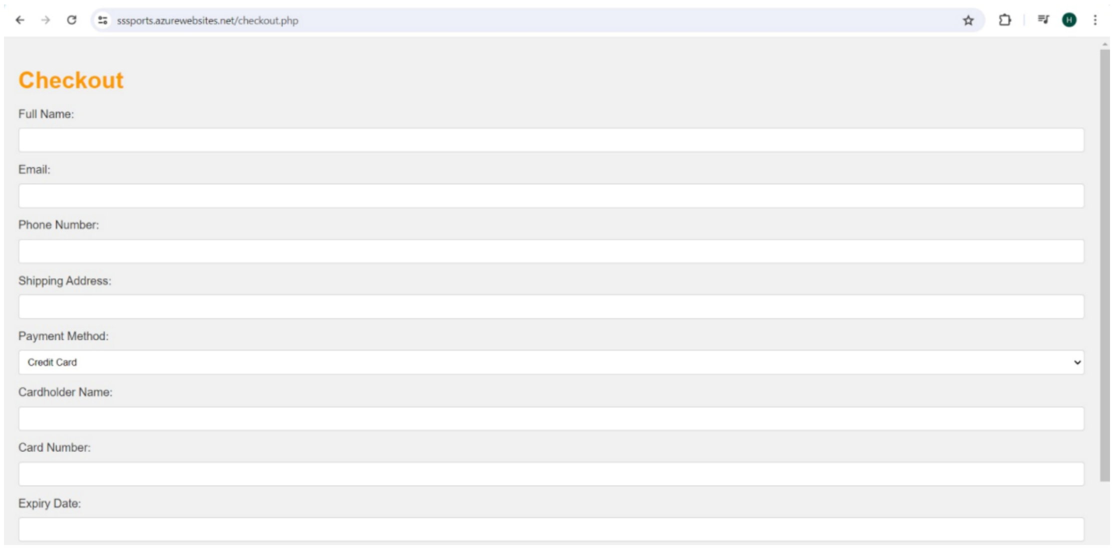
Cart page:



Your Cart:

Cricket Bat - $39.99 x 2 Remove
Volleyball - $19.99 x 1 Remove
Basketball - $24.99 x 1 Remove
Football - $29.99 x 3 Remove
Subtotal: $214.93 | Tax (4%): $8.60 | Total: $223.53
Checkout

Payment page:


sssports.azurewebsites.net/checkout.php

Checkout

Full Name:

Email:

Phone Number:

Shipping Address:

Payment Method:

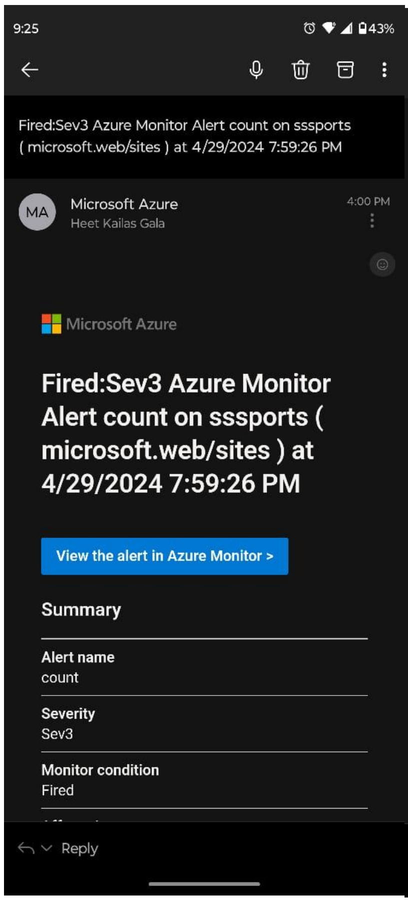
Credit Card

Cardholder Name:

Card Number:

Expiry Date:

Logs custom email alert:


9:25 43%

Fired:Sev3 Azure Monitor Alert count on sssports (microsoft.web/sites) at 4/29/2024 7:59:26 PM

Microsoft Azure
Heet Kailas Gala
4:00 PM

Microsoft Azure

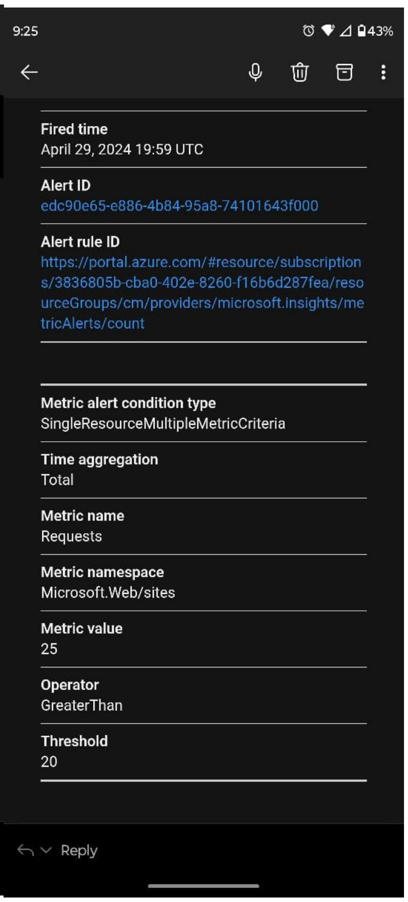
Fired:Sev3 Azure Monitor Alert count on sssports (microsoft.web/sites) at 4/29/2024 7:59:26 PM

View the alert in Azure Monitor >

Summary

Alert name	count
Severity	Sev3
Monitor condition	Fired

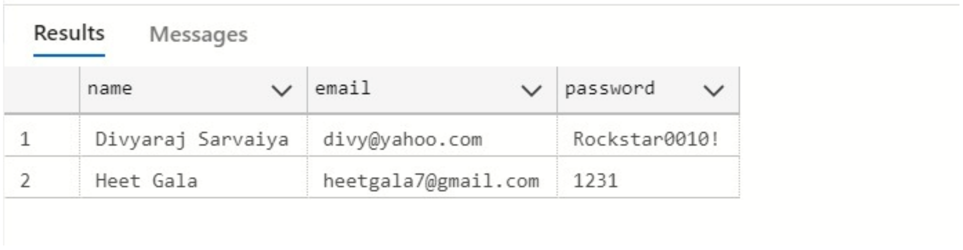
Reply


9:25 43%

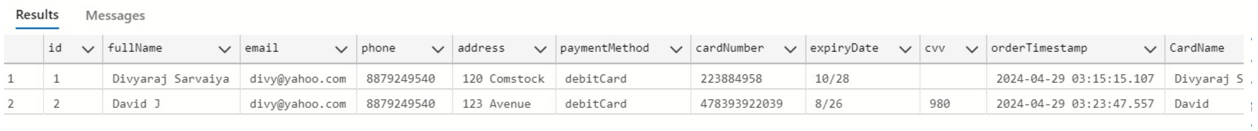
Fired time	April 29, 2024 19:59 UTC
Alert ID	edc90e65-e886-4b84-95a8-74101643f000
Alert rule ID	https://portal.azure.com/#resource/subscriptions/3836805b-cba0-402e-8260-f16b6d287fea/resourceGroups/cm/providers/microsoft.insights/metricAlerts/count
Metric alert condition type	SingleResourceMultipleMetricCriteria
Time aggregation	Total
Metric name	Requests
Metric namespace	Microsoft.Web/sites
Metric value	25
Operator	GreaterThan
Threshold	20

Reply

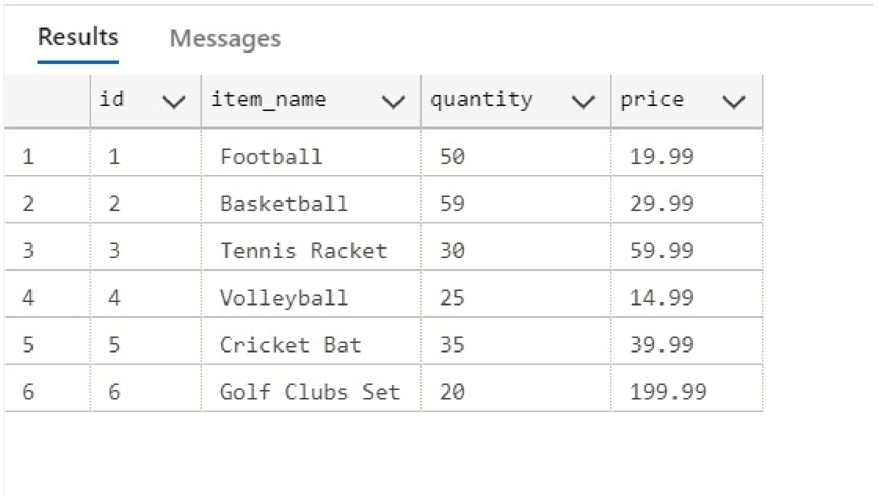
Data tables:


Results Messages

	name	email	password
1	Divyaraj Sarvaiya	divy@yahoo.com	Rockstar0010!
2	Heet Gala	heetgala7@gmail.com	1231


Results Messages

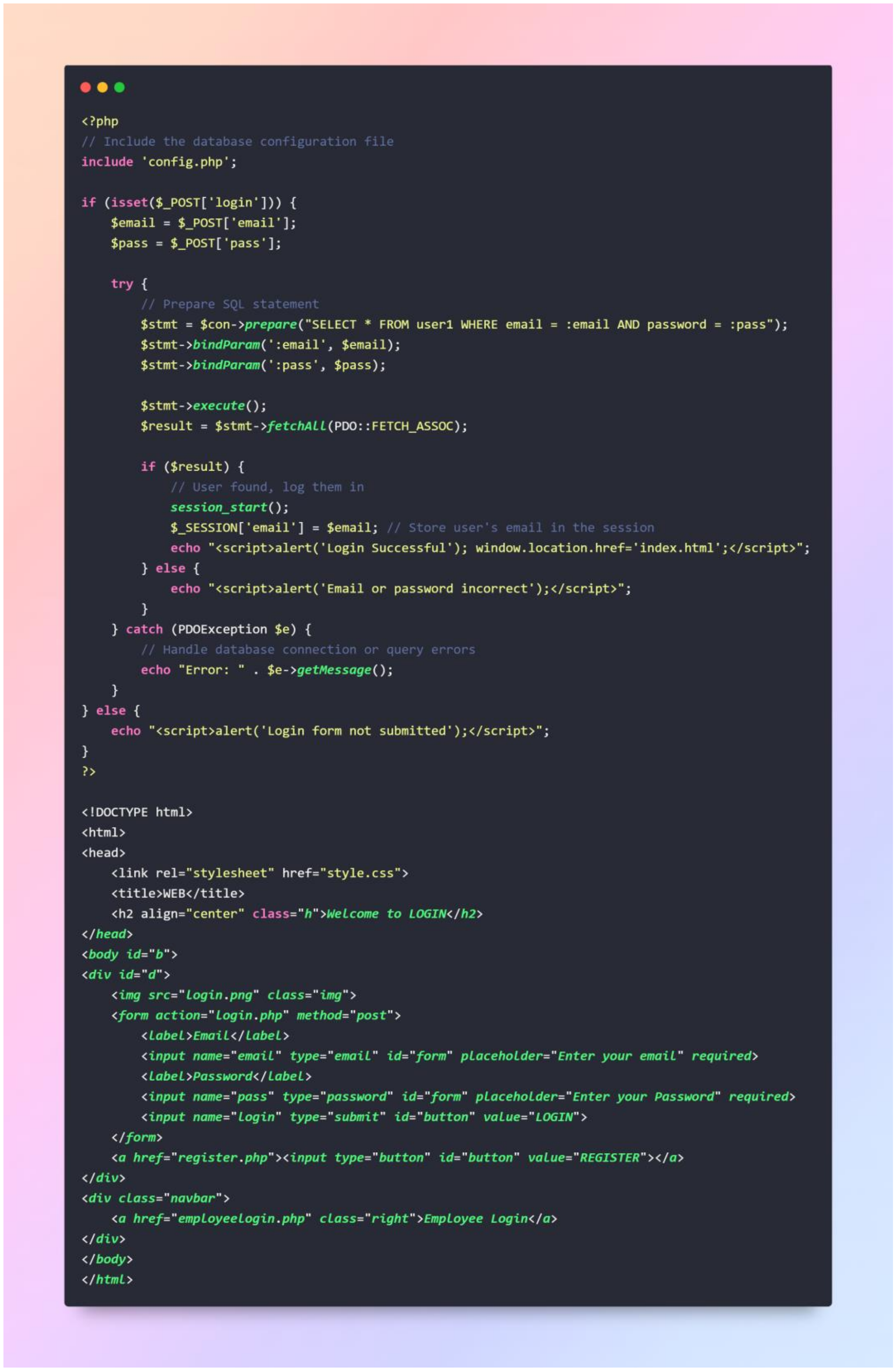
	id	fullName	email	phone	address	paymentMethod	cardNumber	expiryDate	cvv	orderTimestamp	CardName
1	1	Divyaraj Sarvaiya	divy@yahoo.com	8879249540	120 Comstock	debitCard	223884958	10/28		2024-04-29 03:15:15.107	Divyaraj S
2	2	David J	divy@yahoo.com	8879249540	123 Avenue	debitCard	478393922039	8/26	980	2024-04-29 03:23:47.557	David


Results Messages

	id	item_name	quantity	price
1	1	Football	50	19.99
2	2	Basketball	59	29.99
3	3	Tennis Racket	30	59.99
4	4	Volleyball	25	14.99
5	5	Cricket Bat	35	39.99
6	6	Golf Clubs Set	20	199.99

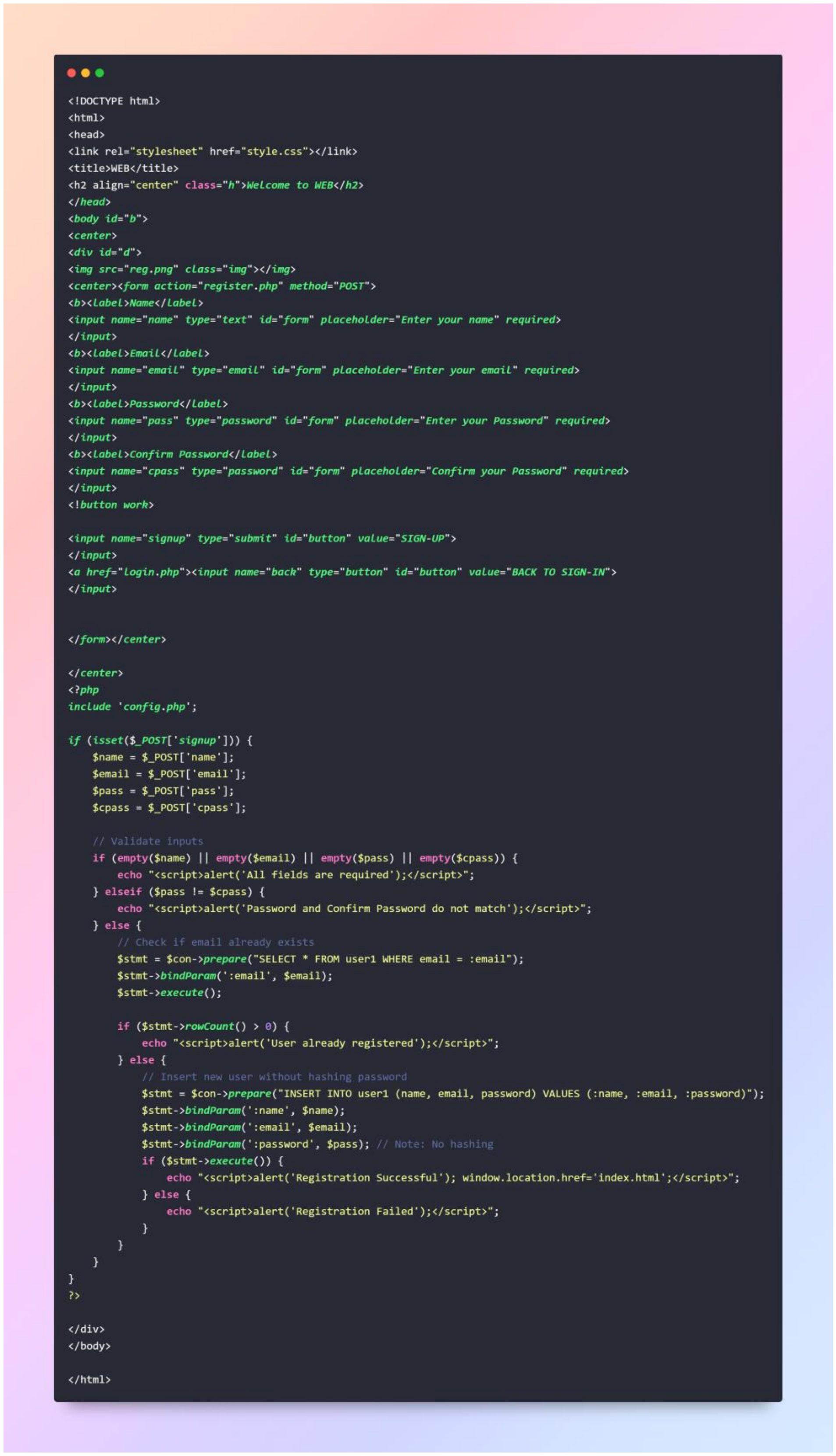
Scripts / Code:

Login file:



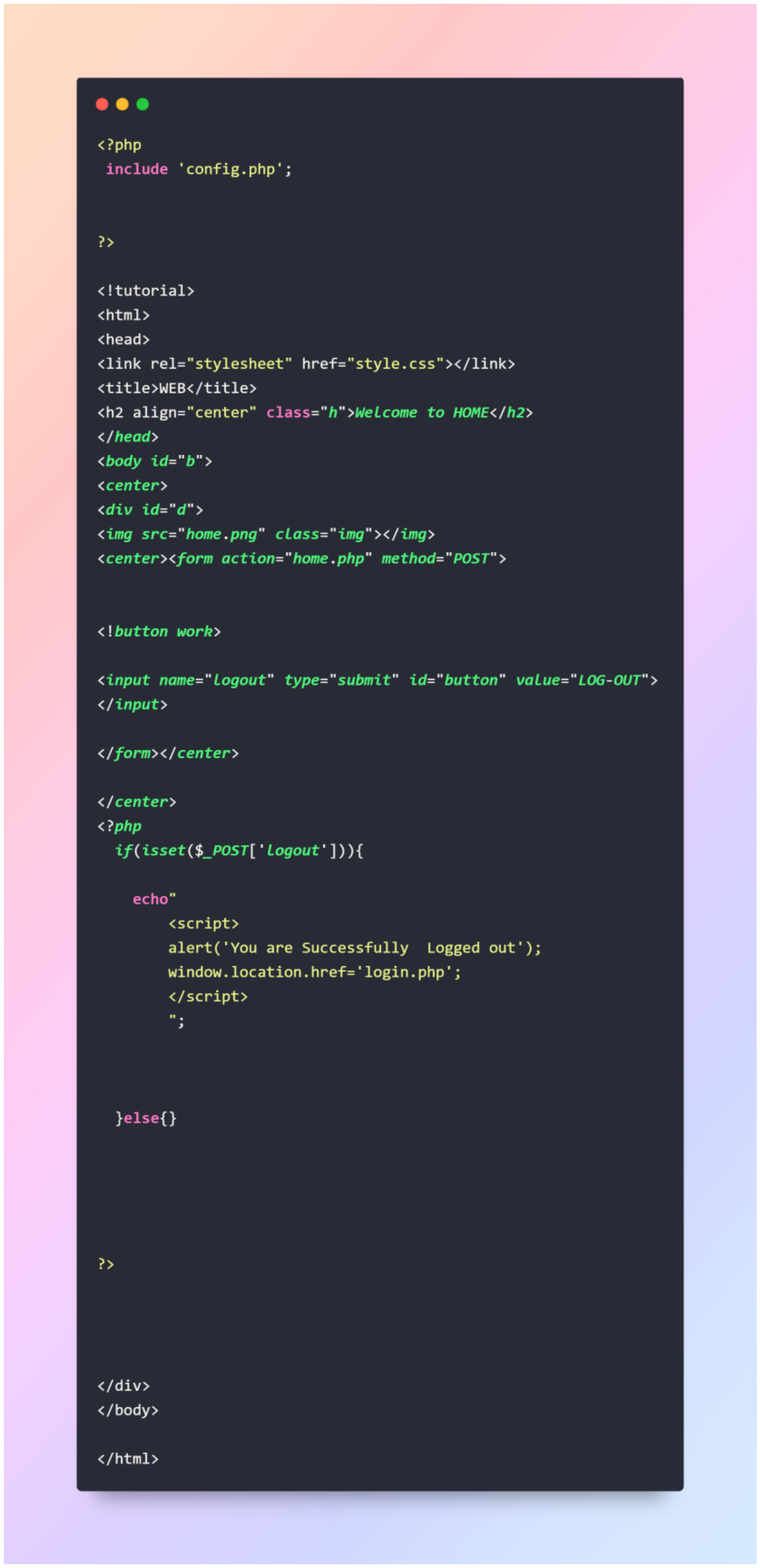
```
<?php
// Include the database configuration file
include 'config.php';

if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $pass = $_POST['pass'];

    try {
        // Prepare SQL statement
        $stmt = $con->prepare("SELECT * FROM user1 WHERE email = :email AND password = :pass");
        $stmt->bindParam(':email', $email);
        $stmt->bindParam(':pass', $pass);

        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);

        if ($result) {
            // User found, log them in
            session_start();
            $_SESSION['email'] = $email; // Store user's email in the session
            echo "<script>alert('Login Successful'); window.location.href='index.html';</script>";
        } else {
            echo "<script>alert('Email or password incorrect');</script>";
        }
    } catch (PDOException $e) {
        // Handle database connection or query errors
        echo "Error: " . $e->getMessage();
    }
} else {
    echo "<script>alert('Login form not submitted');</script>";
}
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
    <title>WEB</title>
    <h2 align="center" class="h">Welcome to LOGIN</h2>
</head>
<body id="b">
<div id="d">
    <img src="login.png" class="img">
    <form action="login.php" method="post">
        <label>Email</label>
        <input name="email" type="email" id="form" placeholder="Enter your email" required>
        <label>Password</label>
        <input name="pass" type="password" id="form" placeholder="Enter your Password" required>
        <input name="login" type="submit" id="button" value="LOGIN">
    </form>
    <a href="register.php"><input type="button" id="button" value="REGISTER"></a>
</div>
<div class="navbar">
    <a href="employeelogin.php" class="right">Employee Login</a>
</div>
</body>
</html>
```

Register file:



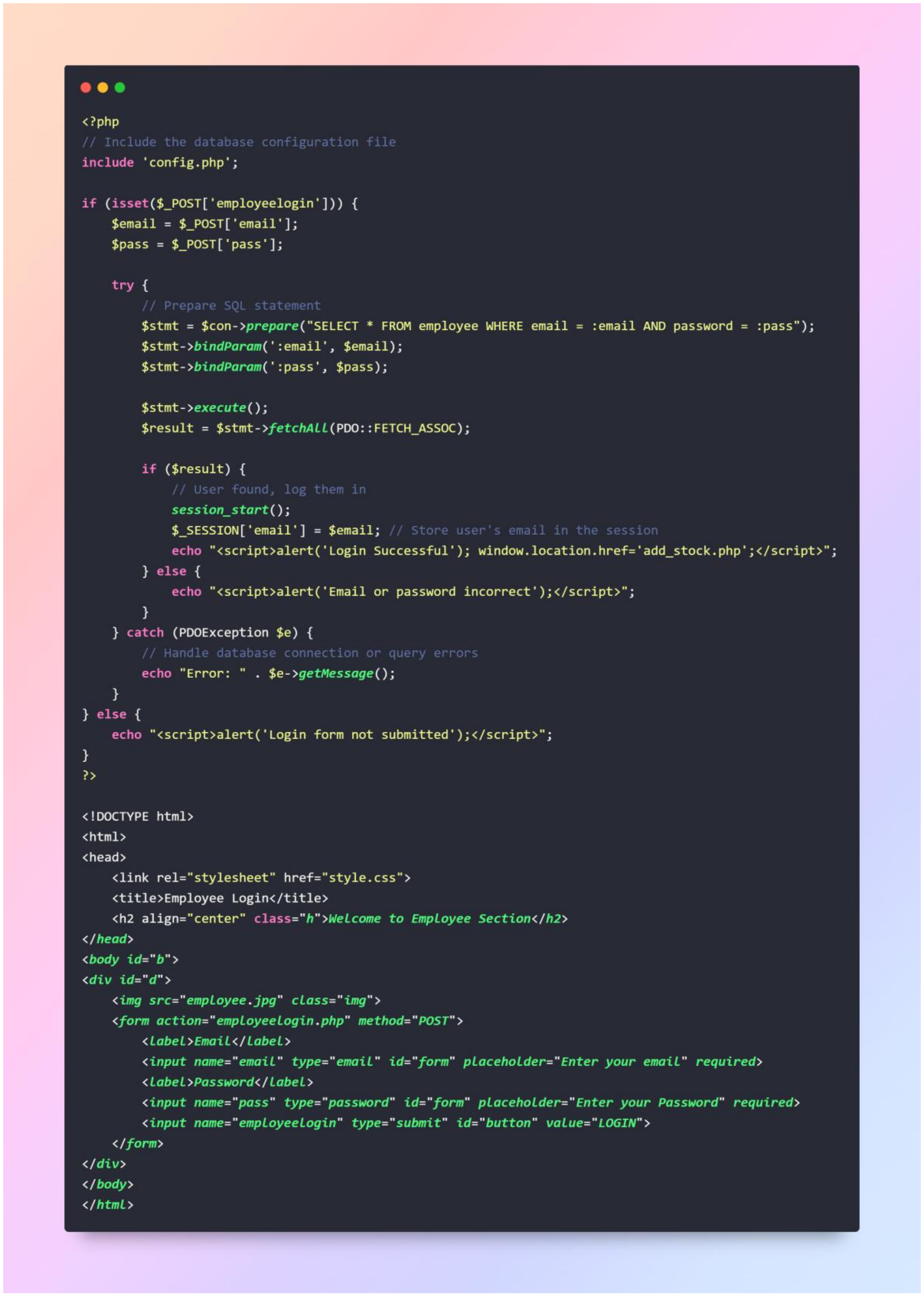
```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="style.css"></link>
<title>WEB</title>
<h2 align="center" class="h">Welcome to WEB</h2>
</head>
<body id="b">
<center>
<div id="d">
<img src="reg.png" class="img"></img>
<center><form action="register.php" method="POST">
<b><label>Name</label>
<input name="name" type="text" id="form" placeholder="Enter your name" required>
</input>
<b><label>Email</label>
<input name="email" type="email" id="form" placeholder="Enter your email" required>
</input>
<b><label>Password</label>
<input name="pass" type="password" id="form" placeholder="Enter your Password" required>
</input>
<b><label>Confirm Password</label>
<input name="cpass" type="password" id="form" placeholder="Confirm your Password" required>
</input>
<!button work>

<input name="signup" type="submit" id="button" value="SIGN-UP">
</input>
<a href="login.php"><input name="back" type="button" id="button" value="BACK TO SIGN-IN">
</input>


</form></center>

</center>
<?php
include 'config.php';

if (isset($_POST['signup'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $pass = $_POST['pass'];
    $cpass = $_POST['cpass'];

    // Validate inputs
    if (empty($name) || empty($email) || empty($pass) || empty($cpass)) {
        echo "<script>alert('All fields are required');</script>";
    } elseif ($pass != $cpass) {
        echo "<script>alert('Password and Confirm Password do not match');</script>";
    } else {
        // Check if email already exists
        $stmt = $con->prepare("SELECT * FROM user1 WHERE email = :email");
        $stmt->bindParam(':email', $email);
        $stmt->execute();

        if ($stmt->rowCount() > 0) {
            echo "<script>alert('User already registered');</script>";
        } else {
            // Insert new user without hashing password
            $stmt = $con->prepare("INSERT INTO user1 (name, email, password) VALUES (:name, :email, :password)");
            $stmt->bindParam(':name', $name);
            $stmt->bindParam(':email', $email);
            $stmt->bindParam(':password', $pass); // Note: No hashing
            if ($stmt->execute()) {
                echo "<script>alert('Registration Successful'); window.location.href='index.html';</script>";
            } else {
                echo "<script>alert('Registration Failed');</script>";
            }
        }
    }
}
?>

</div>
</body>

</html>
```

Home file:



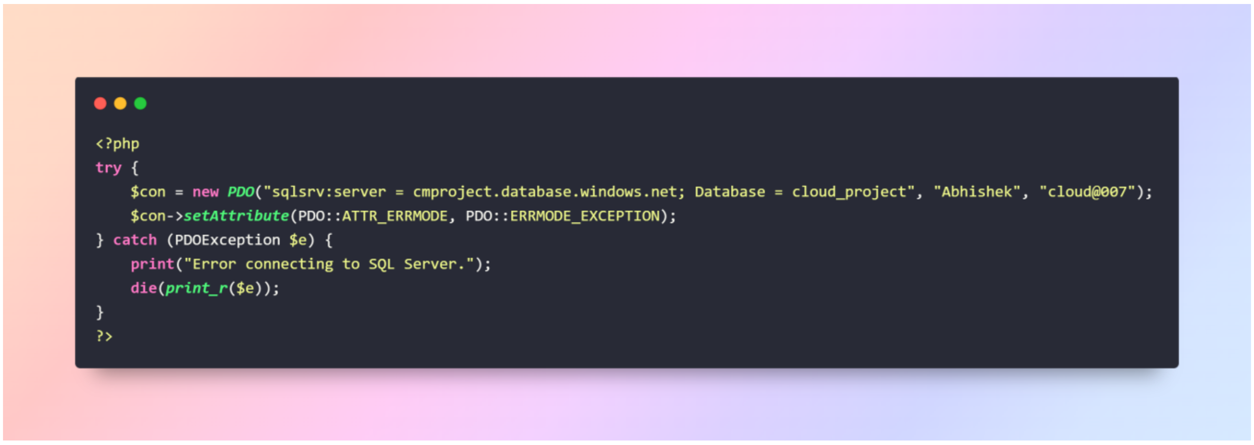
```
<?php
 include 'config.php';


?>

<!tutorial>
<html>
<head>
<link rel="stylesheet" href="style.css"></link>
<title>WEB</title>
<h2 align="center" class="h">Welcome to HOME</h2>
</head>
<body id="b">
<center>
<div id="d">
<img src="home.png" class="img"></img>
<center><form action="home.php" method="POST">


<!button work>

<input name="logout" type="submit" id="button" value="LOG-OUT">
</input>

</form></center>

</center>
<?php
  if(isset($_POST['logout'])){

    echo"
        <script>
        alert('You are Successfully  Logged out');
        window.location.href='login.php';
        </script>
        ";



  }else{}






?>






</div>
</body>

</html>
```

Employee login:



```
<?php
// Include the database configuration file
include 'config.php';

if (isset($_POST['employeelogin'])) {
    $email = $_POST['email'];
    $pass = $_POST['pass'];

    try {
        // Prepare SQL statement
        $stmt = $con->prepare("SELECT * FROM employee WHERE email = :email AND password = :pass");
        $stmt->bindParam(':email', $email);
        $stmt->bindParam(':pass', $pass);

        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);

        if ($result) {
            // User found, log them in
            session_start();
            $_SESSION['email'] = $email; // Store user's email in the session
            echo "<script>alert('Login Successful'); window.location.href='add_stock.php';</script>";
        } else {
            echo "<script>alert('Email or password incorrect');</script>";
        }
    } catch (PDOException $e) {
        // Handle database connection or query errors
        echo "Error: " . $e->getMessage();
    }
} else {
    echo "<script>alert('Login form not submitted');</script>";
}
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
    <title>Employee Login</title>
    <h2 align="center" class="h">Welcome to Employee Section</h2>
</head>
<body id="b">
<div id="d">
    <img src="employee.jpg" class="img">
    <form action="employeelogin.php" method="POST">
        <label>Email</label>
        <input name="email" type="email" id="form" placeholder="Enter your email" required>
        <label>Password</label>
        <input name="pass" type="password" id="form" placeholder="Enter your Password" required>
        <input name="employeelogin" type="submit" id="button" value="LOGIN">
    </form>
</div>
</body>
</html>
```

Config file:



```php
<?php
try {
    $con = new PDO("sqlsrv:server = cmproject.database.windows.net; Database = cloud_project", "Abhishek", "cloud@007");
    $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    print("Error connecting to SQL Server.");
    die(print_r($e));
}
?>
```