**DIT1012 INTERNET BASED PROGRAMMING – TAKE AWAY ASSIGNMENT**

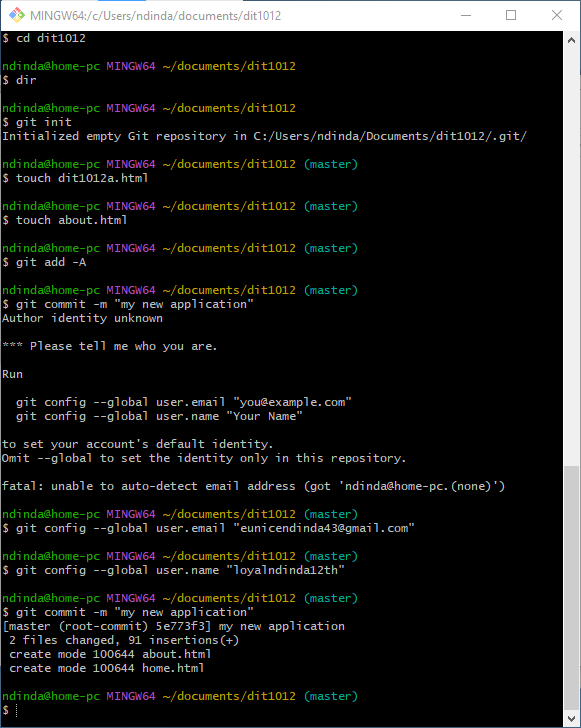
Eunice Ndinda

Department

KIRIRI WOMENS UNIVERSITY OF SCIENCE AND TECHNOLOGY

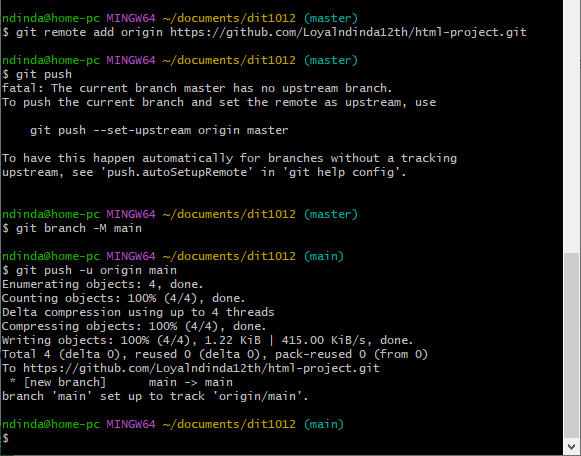
**Section 1: Git and Github**

1. Git basics



1. Using Github

Git repository address: <https://github.com/Loyalndinda12th/html-project.git>



**Section 2 HTML and CSS**

1. HTML Structure

Index/Home Page

<!DOCTYPE html>

<!-- Eunice Ndinda -->

<html lang="en" dir="ltr">

  <head>

    <meta charset="UTF-8">

    <meta name="description" content="">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> Group Assignment - DIT1012 </title>

    <link rel="stylesheet" href="style.css">

    <script src="https://kit.fontawesome.com/a076d05399.js"></script>

   </head>

<body>

  <nav>

    <div class="menu">

      <input type="checkbox" id="check">

      <div class="logo"><a href="#">WorldWide Tech Solutions</a></div>

        <ul>

          <label class="btn cancel" for="check"><i class="fas fa-times"></i></label>

          <li><a href="#">Home</a></li>

          <li><a href="/about.html">About</a></li>

          <li><a href="/contact.html">Contact</a></li>

        </ul>

        <label for="check" class="btn bars"><i class="fas fa-bars"></i></label>

      </div>

    </div>

  </nav>

  <div class="center">

    <img src="assets/image.webp" class="centered"/>

    <div class="upper">

      Welcome to World Wide Tech Solutions

    </div>

    <div class="lower">

      Home to your software solutions

    </div>

  </div>

  <div class="footer">

    DIT1012 Group Assignment

  </div>

</body>

</html>

About Page

<!DOCTYPE html>

<!-- Eunice Ndinda -->

<html lang="en" dir="ltr">

  <head>

    <meta charset="UTF-8">

    <meta name="description" content="">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> Group Assignment - DIT1012 </title>

    <link rel="stylesheet" href="style.css">

    <script src="https://kit.fontawesome.com/a076d05399.js"></script>

   </head>

<body>

  <nav>

    <div class="menu">

      <input type="checkbox" id="check">

      <div class="logo"><a href="#">WorldWide Tech Solutions</a></div>

        <ul>

          <label class="btn cancel" for="check"><i class="fas fa-times"></i></label>

          <li><a href="/index.html">Home</a></li>

          <li><a href="#", color="blue">About</a></li>

          <li><a href="/contact.html">Contact</a></li>

        </ul>

        <label for="check" class="btn bars"><i class="fas fa-bars"></i></label>

      </div>

    </div>

  </nav>

  <div class="center">

    <div class="upper">

      About Us

    </div>

    <div class="lower">

        <h1>Company Overview</h1>

        <p>

            WW Tech's flagship product is a lightning-fast,

            secure, and privacy-focused web browser that delivers

            an unparalleled user experience.

        </p>

    </div>

  </div>

  <div class="footer">

    DIT1012 Group Assignment

  </div>

</body>

</html>

Contact Page

<!DOCTYPE html>

<!-- Eunice Ndinda -->

<html lang="en" dir="ltr">

  <head>

    <meta charset="UTF-8">

    <meta name="description" content="">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> Group Assignment - DIT1012 </title>

    <link rel="stylesheet" href="style.css">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <script src="https://kit.fontawesome.com/a076d05399.js"></script>

   </head>

<body>

  <nav>

    <div class="menu">

      <input type="checkbox" id="check">

      <div class="logo"><a href="#">WorldWide Tech Solutions</a></div>

        <ul>

          <label class="btn cancel" for="check"><i class="fas fa-times"></i></label>

          <li><a href="/index.html">Home</a></li>

          <li><a href="/about.html", color="blue">About</a></li>

          <li><a href="#">Contact</a></li>

        </ul>

        <label for="check" class="btn bars"><i class="fas fa-bars"></i></label>

      </div>

    </div>

  </nav>

  <div class="center">

    <div class="container mt-5">

        <h1>Contact Us</h1>

        <form class="contact-form">

            <div class="form-group">

                <label for="name">Name:</label>

                <input type="text" id="name" class="form-control" placeholder="Enter your name" required>

            </div>

            <div class="form-group">

                <label for="email">Email:</label>

                <input type="email" id="email" class="form-control" placeholder="Enter your email" required>

            </div>

            <div class="form-group">

                <label for="message">Message:</label>

                <textarea id="message" class="form-control" rows="4" placeholder="Type your message here" required></textarea>

            </div>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

  </div>

  <div class="footer">

    DIT1012 Group Assignment

  </div>

</body>

</html>

1. CSS styling Cascading Stylesheets (style.css)

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Poppins',sans-serif;

}

body{

  background: #ecf0f3;

}

nav{

  width: 100%;

  padding: 12px 0;

  background: #ecf0f3;

  box-shadow: -3px -3px 7px #ffffff,

             3px 3px 5px #ceced1,

             inset -3px -3px 7px #ffffff,

             inset 3px 3px 5px #ceced1;

}

nav .menu{

  max-width: 1270px;

  margin: auto;

  padding: 0 20px;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.menu .logo a{

  font-size: 28px;

  font-weight: 600;

  text-decoration: none;

  color: #31344b;

}

.menu ul{

  list-style: none;

  display: flex;

}

.menu ul a{

  margin:0 8px;

  text-decoration: none;

  font-size: 18px;

  color: #31344b;

  font-weight: 400;

  display: inline-flex;

  padding: 10px 12px;

 box-shadow: -3px -3px 7px #ffffff,

            3px 3px 5px #ceced1;

  position: relative;

  transition: all 0.3s ease;

}

.menu ul a:hover:before{

  content: '';

  position: absolute;

  top: 0;

  left: 0;

  right: 0;

  bottom: 0;

  box-shadow: inset -3px -3px 7px #ffffff,

              inset 3px 3px 5px #ceced1;

}

.center{

  position: absolute;

  top: 50%;

  left: 50%;

  width: 100%;

  transform: translate(-50%, -50%);

  text-align: center;

}

.upper{

  font-size: 40px;

  font-weight: 600;

  color: #31344b;

}

.lower{

  font-size: 20px;

  font-weight: 600;

  color: #31344b;

  font-style: italic;

}

.menu ul a:hover{

  color: #3498db;

}

nav label.btn{

  color: #31344b;

  font-size: 18px;

  cursor: pointer;

  display: none;

}

nav label.cancel{

  position: absolute;

  top: 25px;

  right: 30px;

}

#check{

  display: none;

}

@media (max-width:940px) {

  .menu ul{

    display: block;

    position: fixed;

    top: 0;

    left: -100%;

    width: 100%;

    max-width: 400px;

    padding-top: 45px;

    height: 100%;

    background: pink;

    box-shadow: 0 5px 10px #b0b0b5;

    z-index: 12;

    transition: all 0.3s ease;

  }

  .menu ul a{

   display: block;

   font-size: 23px;

   width: 100%;

   margin-top: 30px;

   box-shadow: none;

   text-align: center;

  }

  .menu ul a:hover:before{

    box-shadow: none;

  }

  nav label.bars{

    display: block;

  }

  #check:checked ~ label.bars{

    display: none;

  }

  #check:checked ~ ul label.cancel{

    display: block;

  }

  #check:checked ~ ul{

    left: 0;

  }

}

.centered {

    display: block;

    margin-left: auto;

    margin-right: auto;

    width: 50%;

    height: 50%;

}

.footer {

    padding: 1px;

    position: fixed;

    left: 0;

    bottom: 0;

    width: 100%;

    height: 1cm;

    background-color: #a9afff;

    color: black;

    text-align: center;

  }

.contact-form {

    max-width: 400px;

    margin: 0 auto;

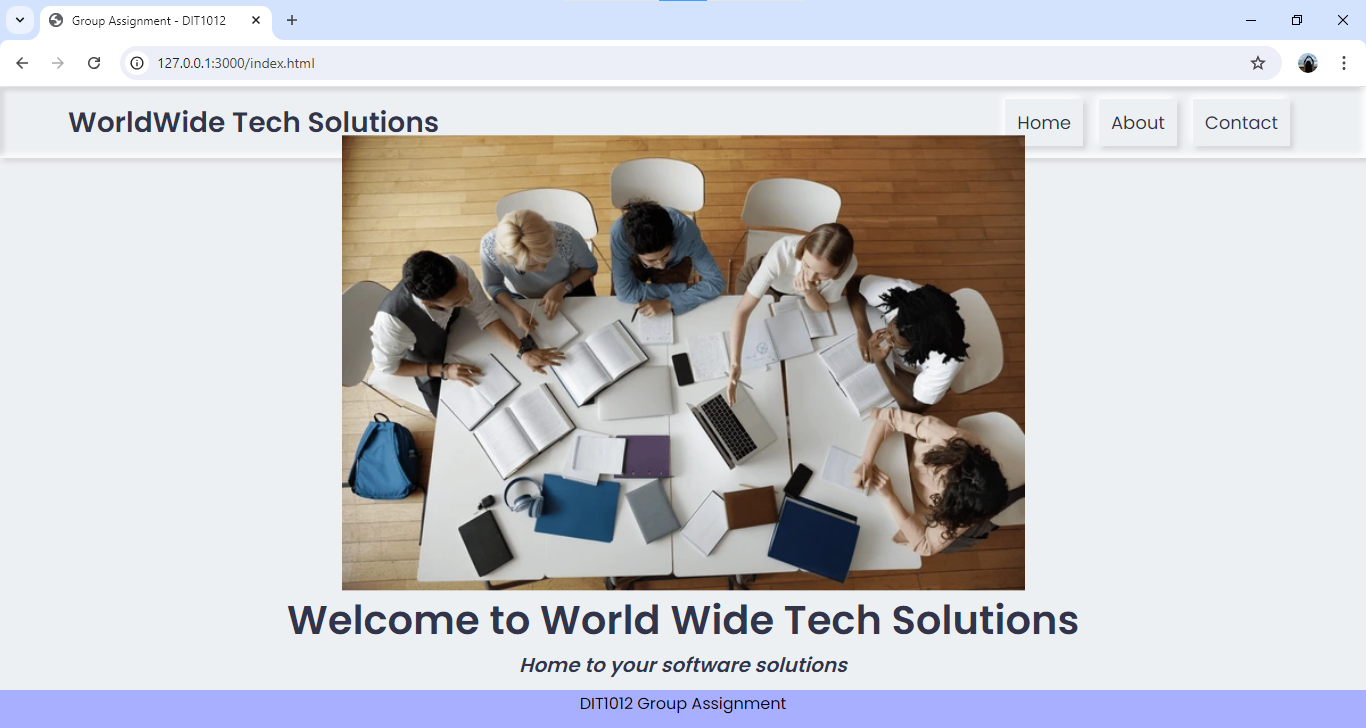
}

.form-group label {

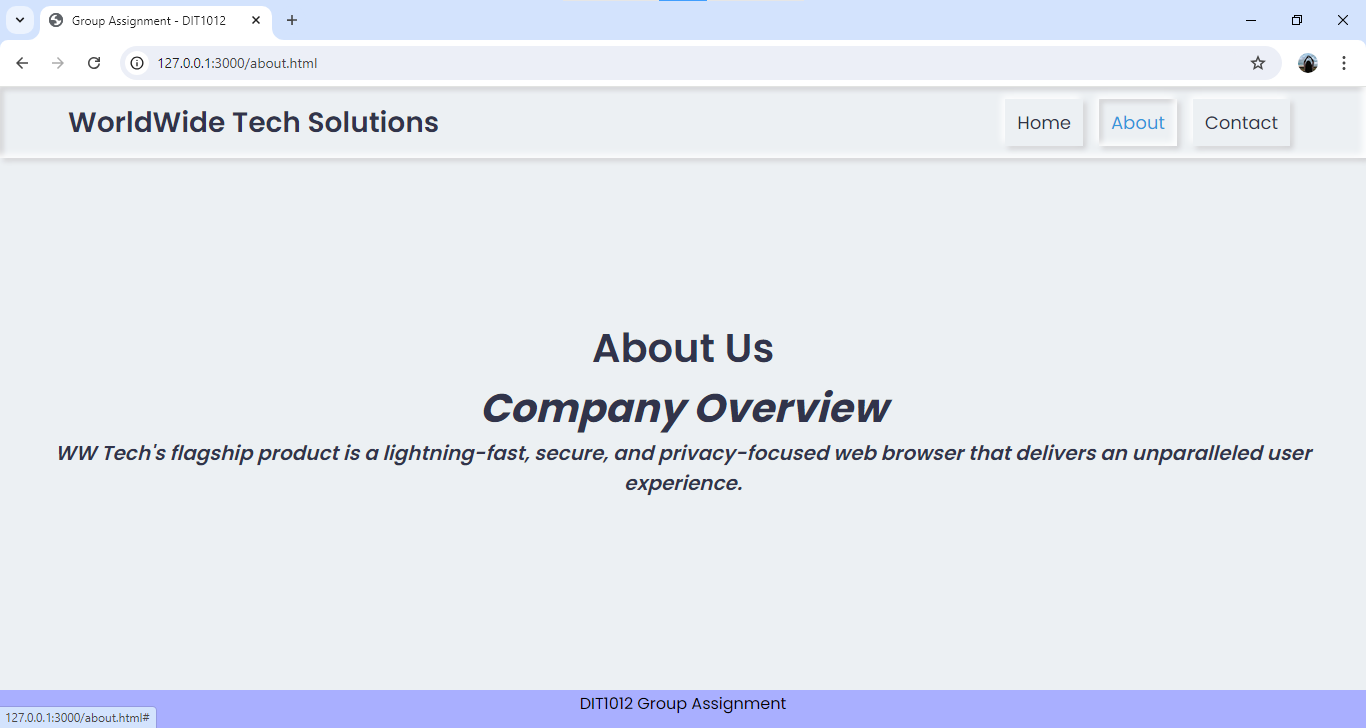
    font-weight: bold;

}

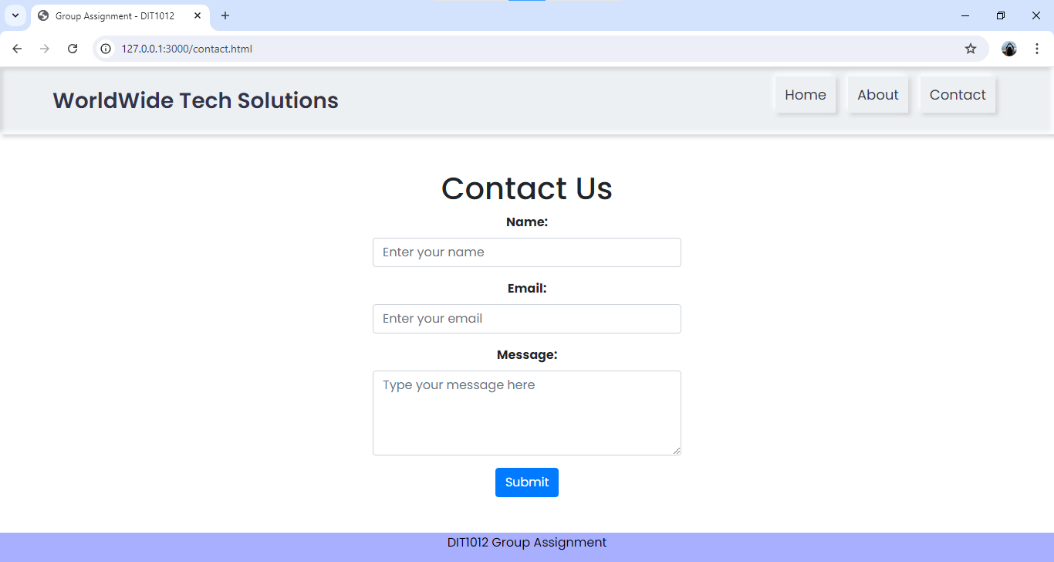
Home Page:



About Page



Contact page



**Section 3: Web Programming**

Javascript interactivity

Script.js

const contactForm = document.querySelector('.contact-form');

  contactForm.addEventListener('submit', (event) => {

    event.preventDefault(); // Prevent default form submission

    // Get form data

    const name = document.getElementById('name').value;

    const email = document.getElementById('email').value;

    const message = document.getElementById('message').value;

    // Basic validation

    if (name.trim() === '') {

        alert('Please enter your name.');

        return; // Stop further processing

    }

    if (!validateEmail(email)) {

        alert('Please enter a valid email address.');

    return;

}

if (message.trim() === '') {

    alert('Please enter a message.');

    return;

  }

  // Create a simple object to hold the data

  const formData = {

    name: name,

    email: email,

    message: message

  };

  alert(`Thank you for your message, ${name}! We'll get back to you soon.`);

  // Clear the form fields

  contactForm.reset();

});

// Simple email validation function

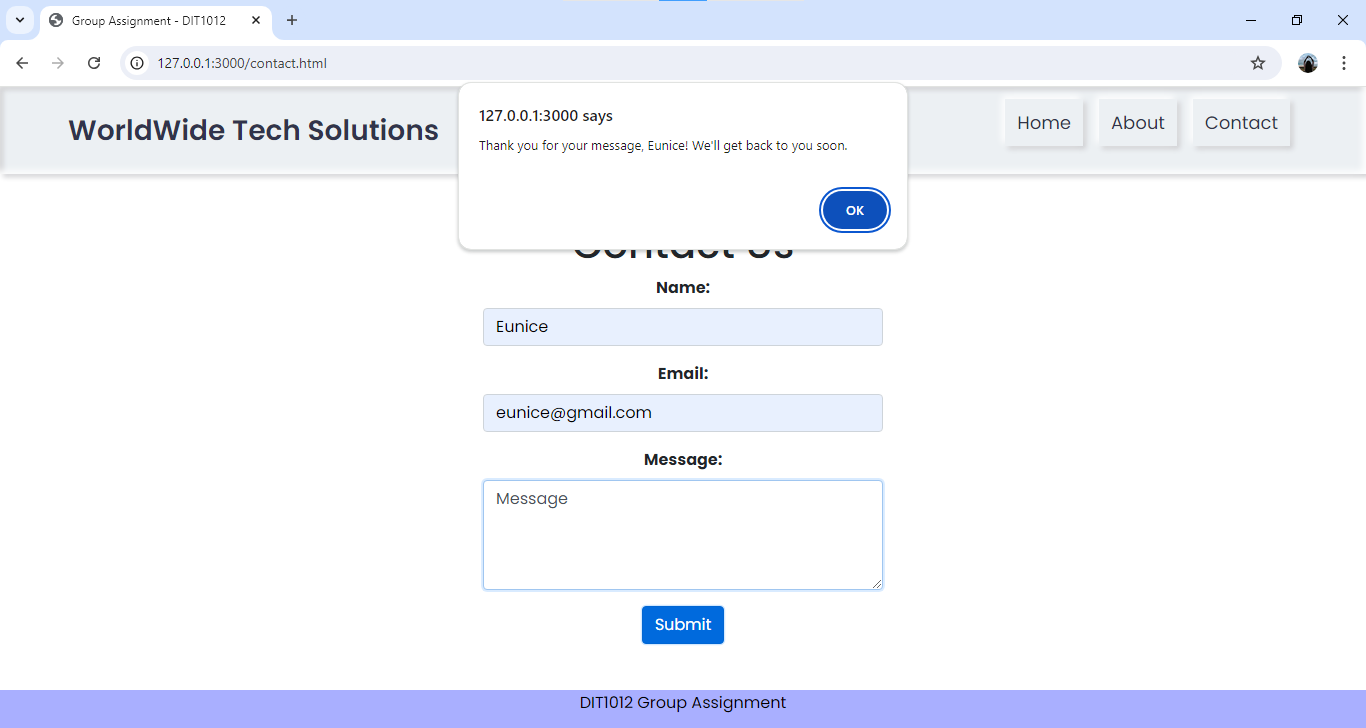
function validateEmail(email) {

  // You can use a more robust regex if needed

  const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  return re.test(String(email).toLowerCase());

}



**Web security best practices**

Web applications have been the actual backbones of businesses and personal interactions in this modern digital age. Their increasing complexity, however, has made them potential playgrounds for cyberattacks. Therefore, robust measures in web security are very essential to protect sensitive data and user trust.

Input validation is the most basic practice. It deals with meticulously checking all data given by the user to prevent malignant input. Another very common vulnerability is SQL injection. This could be reduced by the use of parameterized queries and input sanitation. In so doing, an organization will help secure its database from unwanted access and changes by preventing unauthorized SQL commands.

Another critical aspect is secure authentication and authorization. This involves adherence to a good password policy, enforcing multi-factor authentication, and usage of role-based access controls. It seeks to ensure that sensitive information is accessed only by authorized persons. Coupled with auditing inactive user accounts regularly by revoking them, this helps reduce this risk of unauthorized access.

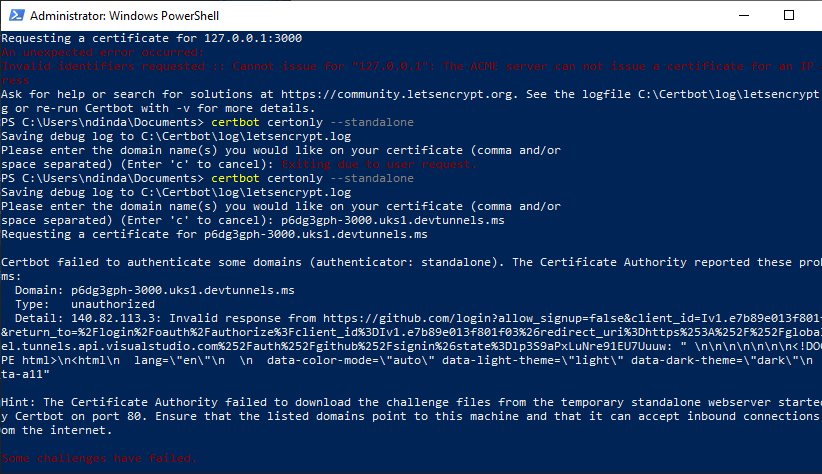
First and foremost, data protection requires that data be encrypted. Only the most secure algorithms for encrypting data at rest and in transit should be used, with the keys updated frequently. Key management practices should also be implemented.

Security testing and vulnerability assessments should be conducted regularly. Penetration testing allows one to realize the weak points in the defenses of an application so that its remediation could be done in time. A web application firewall can add another layer of protection by filtering and observing incoming traffic.

Basically, the key thing is to update the security practices regularly and to patch up vulnerabilities without wasting a single moment. Cybercriminals never stop reforming their tactics and find the minutest weaknesses in the system. Therefore, by adopting a proactive approach toward security, getting updated about emerging threats and vulnerabilities will help an organization stay ahead of any potential attacks.

Organizationally, therefore, these best practices can help keep the risk of web application attacks at bay. Only with a comprehensive security strategy will input validation, secure authentication, data encryption, testing, and updating serve to protect sensitive information and user trust within the digital environment.

**Section 4: Web Security**

****

**Section 5: Emerging Trends in Internet-Based Programming**

Due to the development and changes in technology and the needs of society, the scene of internet-based programming is changing rapidly. This paper looks into three emerging trends that modify the scenery: increasing serverless architecture, growing importance of WebAssembly (Wasm), and rising usage of AI-driven development tools.

Serverless architecture is changing the way developers used to build and deploy applications. Contrary to the name, serverless computing does not eliminate servers; on the other hand, it abstracts server management away from the developer. This model allows developers to focus solely on writing code without worrying about the underlying infrastructure.

Probably one of the most well-known examples of serverless computing is AWS Lambda, offered by Amazon Web Services (Amazon Web Services, nd.). With AWS Lambda, code will be executed in response to events like data changes in an S3 bucket or a table update in DynamoDB, all without the need and hassle of provisioning a server or managing it. Its serverless model scales automatically, only charges for actual use, and makes development much easier.

Another example is Azure Functions from Microsoft, which enables developers to write event-driven code in a serverless environment as well. This architecture is particularly useful for applications whose workload periodically goes up and down, like real-time data processing or unpredictable user traffic. This model decreases costs and operational overhead, enables quicker deployment of updates, and reduces the number of infrastructural concerns to be maintained by the developer.

WebAssembly is the new technology rewriting the rules in web development. Wasm makes it possible to execute high-performance code on the web. Compared to an interpreted JavaScript language, WebAssembly is a low-level binary format directly executable by the browser's virtual machine, attaining near-native performance and efficiency (Haas et al., 2020). WebAssembly's potential comes alive in the number of languages with which it integrates. For example, code in languages like C++ or Rust can be compiled into WebAssembly modules. This makes it possible to execute high-performance applications, including games, complex simulations, and data-intensive operations, directly in a browser. Take the case of Unity, a game engine that began compiling games into WebAssembly, realizing faster load times and smoother performance.

Another application domain of WebAssembly is in facilitating new web applications that require high computational performance. For instance, Figma is a Web-based design tool that leverages WebAssembly to perform computationally-intensive image processing within a browser, which is hardly doable with JavaScript only (Unity Technologies, 2023). There are also prominent currents of progress, like in cryptography and scientific computing, where performance gains are significant.

Artificial intelligence is already being embedded within development tools to enhance the productivity of a programmer by way of automated tasks, such as code generation, bug detection, code optimization, and testing automation. One glaring example of AI-driven development is GitHub Copilot, powered by OpenAI's Codex. It makes suggestions for codes in real-time and auto-completes them, thereby fast-tracking the speed of coding. Copilot uses machine learning models trained on a large corpus of open-source code to provide contextually relevant code snippets and solutions. For instance, a developer who is writing a function to process user input may get suggestions on how he can handle edge cases or implement best practices that will help increase productivity while at the same time reduce errors (Sentry, n.d.; Bugsnag, n.d.).

Another example is TabNine, which is AI-infused code completion serving its services in conjunction with such popular code editors as VSCode and IntelliJ. TabNine applies the most sophisticated machine learning techniques to predict and suggest according to context and existing base of codes possible completions of them, enhancing the experience of coding and efficiency.

Apart from that, AI is also making waves in the areas of automated testing and debugging. Tools like Sentry and Bugsnag do it using artificial intelligence by analyzing application logs against a range of possible problems, providing insights and suggestions for their resolution(Sentry, n.d.; Bugsnag, n.d.). This proactive approach in the detection of errors can enable the developers to resolve issues before they ever reach the end-users, hence ensuring a much higher quality and reliability of the softwares.

The face of internet-based programming changes every day, especially with the emerging trends like serverless architecture, WebAssembly, and AI-driven development tools. Serverless computing makes application deployment easy by abstracting the infrastructure management part, thereby letting a developer focus on writing code and bringing efficiency into it. WebAssembly improves the performance of the web, making high-speed execution of code possible and thereby unleashing a wealth of new opportunities for web applications and computational tasks. Meanwhile, the AI-driven development tools change the coding process: providing wiser completion, detecting bugs and optimization. These developments in trends will most probably impact the future of programming as innovation will bloom and strategically set how applications should be built and maintained in the future. In such a fast-changing technological environment, developers have to keep up with these developments to remain competitive.

References:

1. Amazon Web Services. (n.d.). *AWS Lambda*. Retrieved from <https://aws.amazon.com/lambda/>
2. GitHub. (2023). *GitHub Copilot*. Retrieved from <https://github.com/features/copilot>
3. Haas, A., Holman, M., & Reinders, J. (2020). *WebAssembly: The next big thing in web technology*. Retrieved from <https://webassembly.org>
4. Microsoft. (n.d.). *Azure Functions*. Retrieved from <https://azure.microsoft.com/en-us/services/functions/>
5. Sentry. (n.d.). *Sentry for error tracking*. Retrieved from <https://sentry.io>
6. TabNine. (2023). *TabNine AI code completion*. Retrieved from <https://tabnine.com>