**Part 6**

**Web Development**

|  |  |
| --- | --- |
| **4.1** | **Introduction to Web Development** |
|  | Web development refers to the process of creating and maintaining websites and web applications. It involves a combination of various technologies, programming languages, and tools to build interactive and functional websites that can be accessed over the internet. Web development can be broadly categorized into two main components: front-end development and back-end development.  web development involves building web applications using technologies provided by Microsoft.  ASP.NET Web Forms  ASP.NET MVC (Model-View-Controller)  ASP.NET Web API |
| **4.2** | **Web API project** |
|  | Creating a basic ASP.NET Web API project in the .NET Framework involves the following steps.   * Click on "Create a new project." * Select "ASP.NET Web Application" from the project templates. * Choose the "ASP.NET Web API" template. * Click "Create." * Configure the project settings, including the project name, location, and solution name. * Open the "Controllers" folder. * Right-click and choose "Add" -> "Controller." * Select "Web API 2 Controller - Empty" or "Web API 2 Controller with read/write actions." * Press F5 or use the "Start Debugging" button to run the project. * Open a web browser or a tool like [Postman](https://www.postman.com/) to test your Web API. |
| **4.3** | **Action Method Response (HTTP status code etc.)** |
|  | In ASP.NET Web API, the action methods of your controller can return various types of responses, including different HTTP status codes. The response type depends on the nature of the operation and the desired behavior.  Ok (200):  Return Ok() to indicate a successful request.  Created (201):  Return Created to indicate that a resource has been successfully created.  Bad Request (400):  Return BadRequest to indicate that the request is invalid.  Not Found (404):  Return NotFound to indicate that a resource was not found.  Internal Server Error (500):  Return InternalServerError for unexpected server errors. |
| **4.4** | **Security(CORS)** |
|  | CORS (Cross-Origin Resource Sharing) is a security feature implemented by web browsers that restricts webpages from making requests to a different domain than the one that served the original web page. This policy helps prevent potential security risks such as cross-site request forgery. |
|  | **Authentication , Authorization** |
|  | **Authentication:**  Authentication is the process of verifying the identity of a user, system, or application trying to access a resource. In the context of a web API, it involves validating the credentials of the entity making the request.  **Authorization:**  Authorization is the process of determining whether an authenticated user has the necessary permissions to perform a specific action or access a particular resource. It ensures that users are allowed to perform the requested operations based on their roles or claims. |
|  | **Exception** |
|  | Exception handling in a Web API is crucial for providing a good user experience, logging errors, and ensuring the stability of the application  Exception are the errors that happens at runtime . If any error throw in web api it is translated into httpresponse with status code – 500 –internal server Error  -**way to handle exception:**  1. HttpError  2.HttpResponseException  3.ExceptionFilter  4.ExceptionHandler |
|  | **Use of swagger** |
|  | **API Specification:**  **Purpose**: Swagger is primarily used for designing, documenting, and testing APIs.  **API Specification Format**: It uses the OpenAPI Specification (formerly known as Swagger Specification), which is a standard format to describe RESTful APIs.  **Design and Documentation:** Developers can design and document their APIs in a machine-readable format using Swagger.  **Interactive Documentation:** Swagger provides interactive documentation for APIs, allowing users to understand endpoints, parameters, and responses. |
|  | **Use of Postman** |
|  | **API Testing and Development:**  **Purpose:** Postman is primarily used for API testing, development, and collaboration.  **Request Building:** Developers can use Postman to create and send HTTP requests to API endpoints.  **Environment Variables:** Postman supports the use of environment variables, making it easy to manage different configurations (e.g., development, testing, production). |