**1. List of All Your Projects**

**Project 1:** **Meeseva**  
**Position:** Dot Net Developer  
**Description:** An e-governance project to provide multiple government services through a single platform.  
**Technologies Used:** ASP.NET, C#, SQL Server, WCF

**Project 2:** **E-commerce Platform**  
**Position:** Full Stack Developer  
**Description:** An online platform for shopping and product management.  
**Technologies Used:** ASP.NET MVC, Angular, Entity Framework, SQL Server

**Project 3:** **Healthcare Management System**  
**Position:** Software Developer  
**Description:** A comprehensive system for managing hospital operations, including patient records and billing.  
**Technologies Used:** ASP.NET Core, React, MongoDB

**2. Tell Me About Your Current Project**

**Project Name:** Meeseva  
**Objective:** To provide a unified platform for citizens to access various government services and documents.  
**Responsibilities:** Developing and maintaining modules, ensuring smooth integration with backend services, and client communication.  
**Technologies:** ASP.NET, C#, SQL Server, WCF

**3. Tell Me About the Architectures of Your Project**

**Architecture Type:** Layered Architecture  
**Details:** The project follows a layered architecture with a presentation layer (UI), business logic layer, data access layer, and database. The presentation layer is built using ASP.NET, the business logic layer contains C# classes, and the data access layer uses Entity Framework for database operations.

**4. Tell Me About the Modules in Your Project**

**Module 1:** **User Management**  
**Description:** Handles user authentication, authorization, and profile management.

**Module 2:** **Service Request**  
**Description:** Allows users to request various government services and track their status.

**Module 3:** **Payment Gateway**  
**Description:** Integrates with various payment providers to handle online transactions.

**5. Tell Me About the .NET Features Used in Your Current Project**

* **ASP.NET:** For web application development.
* **Entity Framework:** For ORM and database operations.
* **WCF:** For communication with other systems.
* **LINQ:** For data querying.
* **ASP.NET Identity:** For user authentication and authorization.

**6. Tell Me About Your Role and Responsibility in the Current Project**

**Role:** Dot Net Developer  
**Responsibilities:**

* Developing and maintaining modules.
* Performing code reviews and writing unit tests.
* Collaborating with the team for feature implementation.
* Communicating with clients for requirement gathering and feedback.

**7. Tell Me About What Process Models and Types**

**Process Models:** Waterfall, Agile, Scrum, Kanban  
**Description:**

* **Waterfall:** Sequential design process.
* **Agile:** Iterative and incremental approach.
* **Scrum:** Agile framework with roles, events, and artifacts.
* **Kanban:** Visual management of work using boards and cards.

**8. Tell Me About Which Model You Used in the Current Project**

**Model:** Scrum  
**Details:** We follow the Scrum methodology with 2-week sprints, daily stand-ups, sprint planning, reviews, and retrospectives. This helps in continuous delivery and adapting to changing requirements.

**9. Tell Me About Your SDLC and What Phases Exist in That**

**SDLC Phases:**

* **Requirement Gathering:** Collecting client requirements.
* **Design:** Creating architectural and detailed designs.
* **Development:** Coding and implementation.
* **Testing:** Unit, integration, and acceptance testing.
* **Deployment:** Releasing the application to production.
* **Maintenance:** Ongoing support and updates.

**10. Tell Me About What Kind of Documents Exist and Details About Them**

* **Requirement Specifications:** Outlines project requirements and functionalities.
* **Design Documents:** Includes system architecture, database design, and detailed module design.
* **Test Plans:** Contains test cases, testing strategy, and scope.
* **User Manuals:** Guides for end-users on how to use the system.

**11. How Do You Maintain the Configuration of Your Project?**

**Tool:** TFS (Team Foundation Server)  
**Process:** We use TFS for version control, branching, merging, and managing code repositories. Each feature or bug fix is developed in a separate branch and then merged into the main branch after review.

**12. How Do You Start and Plan Your Work?**

**Approach:** We begin with sprint planning, where tasks are prioritized and assigned. Daily stand-ups help track progress, and JIRA is used for task management and tracking.

**13. What Is the Most Challenging Work in Your Career?**

**Situation:** Integrating a third-party payment gateway with complex API requirements.  
**Action:** I researched and understood the API, created a robust integration plan, and implemented it while ensuring security compliance.  
**Outcome:** Successfully integrated the payment gateway, enabling seamless transactions.

**14. What Is the Toughest Situation You Faced in Development?**

**Situation:** A critical bug was found in the production environment that affected a significant portion of users.  
**Action:** We immediately prioritized the issue, performed a hotfix, and deployed it after thorough testing.  
**Outcome:** The issue was resolved quickly, minimizing downtime and customer impact.

**15. How Often Do You Communicate with the Client?**

**Frequency:** Weekly meetings and as needed for urgent matters.  
**Purpose:** To provide updates, discuss new requirements, and gather feedback.

**16. For What Purposes Do You Communicate with the Client?**

**Purposes:** Requirement gathering, progress updates, demo sessions, issue resolution, and receiving feedback.

**17. Explain the Complete Process Followed for the Development**

**Process:**

* **Requirement Gathering:** Understanding client needs.
* **Design:** Creating a system and module-level design.
* **Development:** Coding based on the design documents.
* **Testing:** Unit testing by developers and QA testing.
* **Deployment:** Deploying to a staging environment, then to production.
* **Maintenance:** Providing support and implementing changes.

**18. What Is the Life Cycle Model Used for Development?**

**Model:** Agile (Scrum)  
**Details:** We use Scrum with sprints, daily stand-ups, sprint reviews, and retrospectives to manage work and deliver incrementally.

**19. How Do You Communicate with Team Members?**

**Methods:** Daily stand-ups, Slack for instant messaging, emails, video calls for detailed discussions.

**20. How Do You Demonstrate Excellent Team Management Skills?**

**Examples:**

* Coordinated team efforts during tight deadlines.
* Managed task delegation and ensured balanced workloads.
* Resolved conflicts and encouraged team collaboration.

**21. If Your Client Requests a Change and Asks for Early Delivery, How Will You Manage?**

**Approach:**

* Assess the impact and feasibility.
* Prioritize tasks and allocate resources accordingly.
* Communicate with the client about any compromises on scope or quality if necessary.
* Ensure the team is on board and prepared to meet the new timeline.

**22. How Will You Gather Requirements, and Where Do You Record Them?**

**Methods:** Client meetings, workshops, and brainstorming sessions.  
**Recording Tools:** Use JIRA for detailed requirement documentation, Confluence for additional notes, and Excel/Word for initial drafts.

**23. What Is the Stage When Code Is Delivered to the Client and He Is Testing It?**

**Stage:** User Acceptance Testing (UAT)  
**Details:** The client tests the system to ensure it meets their requirements. Feedback is gathered, and any issues are addressed before final deployment.

**24. How Do You Handle Change Requests?**

**Process:**

* Evaluate the change request's impact on the project scope, timeline, and cost.
* Prioritize and schedule the changes.
* Implement the changes and test them thoroughly.
* Communicate updates with stakeholders.

**25. How Do You Write Unit Test Cases?**

**Approach:**

* Identify the critical functionalities and edge cases.
* Write test methods using frameworks like MSTest, NUnit, or xUnit.
* Ensure tests are automated, repeatable, and cover all possible scenarios.
* Example: For a login function, write tests for valid credentials, invalid credentials, empty input, etc.