Overview:

Develop a web application that collects customer feedback, analyses the sentiment using Azure AI Services, and displays the results in a user-friendly interface. This application will test back-end and front-end coding skills, API integration, decision-making, problem-solving abilities, and basic knowledge of Microsoft Azure and Azure AI Services.

Architecture Overview:

1. **Azure Static Web App**:
   Azure Static Web Apps provide a scalable, secure, and globally distributed platform for hosting static web apps. It seamlessly integrates with Azure Functions and other Azure services, making it a suitable choice for hosting Blazor WebAssembly applications.

2. **Blazor WebAssembly** (UI):
   Blazor WebAssembly allows you to build interactive and client-side web applications using C# and .NET instead of traditional JavaScript. This provides a consistent development experience for .NET developers and allows for code-sharing between the server and client.

3. **Azure Functions** (Serverless Backend):
   Azure Functions offer a serverless compute service that scales automatically based on demand. Using isolated .NET 8 for Azure Functions ensures compatibility with the latest version of the .NET runtime, providing access to the newest features and improvements.

4. **Azure SQL Database**:
   Azure SQL Database is a fully managed relational database service. It offers high availability, security, and scalability. EF Core, as the Object-Relational Mapping (ORM) framework, simplifies database interactions and provides a clean way to query and manipulate data using C# and make changes using migrations.

Technology Choices:

1. **Blazor WebAssembly:**
   Advantages:
   - Code-sharing between server and client.
   - Familiarity for .NET developers.
   - Allows building rich and interactive UIs using C#.

2. **Azure Functions with .NET 8:**
   Advantages:
   - Serverless architecture for automatic scaling.
   - .NET 8 provides the latest language features and improvements.
   - Well-suited for event-driven and microservices architectures.

3. **Azure SQL Database with EF Core:**
   Advantages:
   - Fully managed relational database service.
   - High availability, security, and scalability.
   - EF Core simplifies data access and provides a clean abstraction over database operations.

4. **Azure Static Web Apps:**
   Advantages:
   - Seamless integration with Azure Functions.
   - Global distribution for low-latency access.
   - Integrated CI/CD, authentication, and authorization features.

Integration:

1. **Blazor WebAssembly with Azure Functions:**

   - Use HTTP triggers in Azure Functions to handle API calls from the Blazor application.
   - Leverage Azure Functions as the serverless backend for handling business logic.

2. **Azure Functions with Azure SQL Database:**

   - Use EF Core to perform database operations within the Azure Functions.
   - Utilize Azure SQL Database connection strings securely stored in Static Web App Enviroment variables.

3. **Azure Static Web Apps with Blazor WebAssembly:**

   - Configure static web app routes to direct API calls to the appropriate Azure Functions.
   - Leverage built-in authentication and authorization features of Azure Static Web Apps for securing the application.

Security Considerations:

1. **Authentication and Authorization:**

   - Utilize Azure AD for authentication and configure role-based access control (RBAC) for authorization.
   - Leverage Azure Static Web Apps authentication features.

Scalability:

1. Auto-Scaling:

   - Azure Functions automatically scales based on demand, ensuring optimal resource utilization.

2. Database Scalability:

   - Azure SQL Database offers scalability options such as serverless and provisioned throughput based on workload requirements.

3. Global Distribution:

   - Azure Static Web Apps provide global distribution, reducing latency for users worldwide.

Conclusion:

The chosen technologies and architecture provide a robust, scalable, and secure solution for building and deploying a modern web application. Blazor WebAssembly, Azure Functions, Azure SQL Database, and Azure Static Web Apps complement each other, allowing for efficient development, seamless integration, and optimal performance. Regularly monitoring and securing the application ensures a reliable user experience.