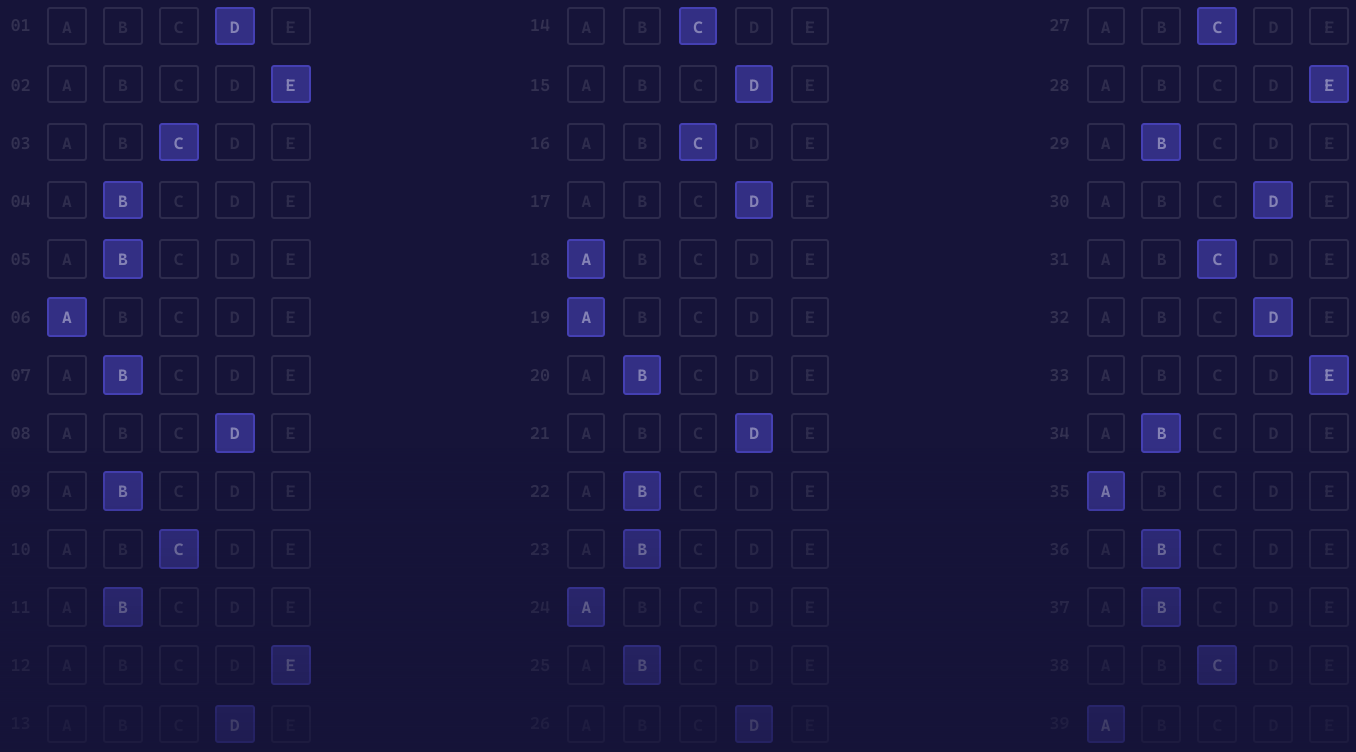
MOMETRIX
SMARTER PREP. BETTER RESULTS.



PRACTICE QUESTIONS

OutSystems

SS11J

Security Specialist O11 (JA)

EDITION	QUESTIONS	FORMAT
Demo	9	Limited Edition PDF

WEB www.mometrix.net

SUPPORT support@mometrix.net

Before you begin

What this is

9 practice questions for OutSystems `SS11J`, each with its answer key and an explanation. Answer a question before you read its key — the explanation is worth more once you have committed.

If something looks wrong

Send us three things and we will check it:

- the exam code, `SS11J`
- the question number
- the email address on your order

Write to **support@mometrix.net**. We reply within one business day.

QUESTION 1 SINGLE CHOICE

An OutSystems Developer is designing the user authentication strategy for a new reactive web application that exposes sensitive financial data. Which of the following is the recommended approach for managing end-user authentication in OutSystems?

A Store user credentials in a local entity in the OutSystems database using a hashing function implemented in a Server Action

B Use the built-in Users application and rely on the OutSystems platform to manage authentication through configured authentication providers

C Create a custom login screen that calls a third-party SOAP web service for credential validation

D Hard-code a single set of administrator credentials in a site property to allow access to all authenticated areas

ANSWER B

WHY OutSystems provides a built-in Users application and a centralized authentication framework that integrates with configured authentication providers (e.g., LDAP, SAML, OIDC). Best practice is to leverage the platform's authentication mechanisms rather than building custom credential stores, which avoids reinventing security controls and benefits from platform-level hardening, password policies, and auditing.

QUESTION 2 SINGLE CHOICE

A developer needs to invoke an external REST API from an OutSystems server action that requires mutual TLS authentication. Where in the OutSystems platform is the client certificate configured to enable this mutual TLS handshake?

A Inside the consumed REST method's URL input parameter as a query string

B In the site's web.config file on each front-end server

C In the Certificate area of the OutSystems environment under the connection settings for the REST API

D By embedding the certificate as a base64 string in a Site Property and decoding it at runtime

ANSWER C

WHY OutSystems manages client certificates used for mutual TLS with external integrations through the platform's environment configuration (the Certificate area associated with the connection). Storing certificates in site properties or web.config bypasses platform controls, while configuring them centrally through the platform keeps certificate lifecycle, rotation and trust store management consistent with OutSystems security best practices.

QUESTION 3 SINGLE CHOICE

Which OutSystems built-in mechanism should a developer use to protect a column that stores personally identifiable information (PII) at rest in an entity, while still allowing the application to query and decrypt the values transparently when needed?

A **Apply the platform's column-level encryption attribute to the entity attribute so the platform handles encryption and decryption**

B Encrypt the column manually using a Forge component that stores the key in a Site Property

C Use the DataProtection API of the underlying web server framework directly in a server action

D Hash the value with a fixed salt in a server action before saving it to the database

ANSWER **A**

WHY OutSystems provides built-in column-level encryption for entity attributes. When applied, the platform transparently encrypts data at rest and decrypts it when the application reads the value, without requiring developers to handle keys in code or build custom cipher logic. This keeps key management under platform control and avoids the pitfalls of manual encryption with key material stored in source artifacts.

QUESTION 4 SINGLE CHOICE

A developer is reviewing a screen preparation that fetches data from an entity based on a user-supplied input from a search box. Which secure coding practice should be applied in OutSystems to prevent injection-style attacks coming through user input on the screen?

A Concatenate the user input directly into an SQL widget using the Execute Query action for performance

B Disable input validation on the search field to reduce server load

C Validate and sanitize input using Input Validation patterns and use parameterized queries through Aggregates or Advanced SQL with bind parameters

D Reflect user input back into a JavaScript expression using the Escape HTML attribute on a local variable

ANSWER C

WHY OutSystems best practice is to treat all user input as untrusted: apply Input Validation and use parameterization via Aggregates or Advanced SQL with bind parameters so values are passed as parameters rather than concatenated into query text. This mitigates SQL injection and similar attacks while still allowing the developer to enforce business rules on the input shape.

QUESTION 5 SINGLE CHOICE

A team is implementing role-based access control in an OutSpaces module. Only members of the Manager role should be able to reach a specific server action that triggers a sensitive refund process. Which is the correct way to enforce this in OutSystems?

A Check the UserId against a hard-coded list inside the action's logic

B Add the Manager role to the action's Role property so only users holding that role can execute it

C Hide the button that calls the action and rely on no one discovering the URL

D Wrap the action call inside an If and Use Client Data action to check the role on the client side

ANSWER B

WHY OutSystems enforces access control server-side based on the Roles assigned to the element (screen, action, or process). Configuring the Role property on the server action guarantees that the platform rejects the request before the logic runs, regardless of any client-side checks. Client-side checks (hiding buttons or client-side if blocks) are not a substitute because they can be bypassed by calling the action directly.

QUESTION 6 SINGLE CHOICE

A Security Specialist runs a Security Analysis scan on an application and the report flags multiple 'Missing required roles' warnings on several server actions. What does this finding indicate?

A The actions are unreachable from any screen and should be deleted

B The actions are granted to the Anonymous role, which makes them publicly invocable without authentication

C The actions have performance issues that need tuning

D The actions are missing a database index on their input parameters

ANSWER B

WHY In OutSystems Security Analysis, a 'Missing required roles' finding on an action typically means the action is reachable by the Anonymous role — i.e., it can be invoked without an authenticated user, which is a common security defect. The remediation is to assign an explicit Role to the action so the platform enforces authentication and authorization before the action runs.

QUESTION 7 SINGLE CHOICE

Which statement correctly describes how HTTPS should be enforced for OutSystems web applications and LifeTime in a production environment?

A HTTPS is automatically enforced at the OutSystems platform layer regardless of front-end server configuration

B HTTPS must be terminated at the front-end web server and the OutServers communicate with the front-end over HTTP by default

C HTTPS must be configured and enforced at the front-end server (e.g., IIS) and applications should be configured to use HTTPS, with HTTP requests redirected

D HTTPS is only required for the LifeTime environment, not for individual applications

ANSWER C

WHY OutSystems relies on the underlying front-end web server (e.g., IIS) to terminate HTTPS. Production deployments should be configured to use HTTPS end-to-end, with HTTP traffic redirected to HTTPS, and applications should be marked/configured to use HTTPS in the environment. LifeTime and individual applications all need transport security in production; the platform does not magically force HTTPS without the front-end being configured for it.

QUESTION 8 SINGLE CHOICE

A Security Specialist needs to grant a third-party contractor temporary access to a single application in the OutSystems environment, but the contractor should not be able to deploy code or change infrastructure. What is the recommended way to grant this least-privilege access?

A Grant them the Environment Administrator role and remove it manually later

B Create an IT user account in LifeTime with the appropriate application-level role only and assign it to the contractor, scoped to the required application

C Share the personal credentials of an existing developer with the contractor

D Add them to the team of every application in the environment so they can audit each one

ANSWER B

WHY OutSystems IT users in LifeTime can be granted scoped roles (e.g., application-level roles such as 'Developer' or 'Viewer' on a specific application) without granting environment-wide privileges. Creating an IT user with only the application-level role required follows least-privilege and avoids sharing personal accounts or granting Environment Administrator rights unnecessarily.

QUESTION 9 MULTIPLE CHOICE

Which of the following are common vulnerability categories reported by the OutSystems Security Analysis tool? (Choose two.)

A Missing required roles on screens and server actions exposed to the Anonymous role

B Use of HTTP (non-secure) URLs in server-side integrations and image sources

C Outdated version of the OutSystems platform that lacks new features

D Slow database queries that exceed a configured response time threshold

ANSWER A · B

WHY OutSystems Security Analysis specifically surfaces security-related issues such as Anonymous access to elements ('Missing required roles') and insecure transport usage such as HTTP URLs in integrations or referenced assets. Performance findings and platform version 'newness' are not part of the security audit scope.