

01	A	B	C	D	E
02	A	B	C	D	E
03	A	B	C	D	E
04	A	B	C	D	E
05	A	B	C	D	E
06	A	B	C	D	E
07	A	B	C	D	E
08	A	B	C	D	E
09	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E

14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E
17	A	B	C	D	E
18	A	B	C	D	E
19	A	B	C	D	E
20	A	B	C	D	E
21	A	B	C	D	E
22	A	B	C	D	E
23	A	B	C	D	E
24	A	B	C	D	E
25	A	B	C	D	E
26	A	B	C	D	E

27	A	B	C	D	E
28	A	B	C	D	E
29	A	B	C	D	E
30	A	B	C	D	E
31	A	B	C	D	E
32	A	B	C	D	E
33	A	B	C	D	E
34	A	B	C	D	E
35	A	B	C	D	E
36	A	B	C	D	E
37	A	B	C	D	E
38	A	B	C	D	E
39	A	B	C	D	E

PRACTICE QUESTIONS

OutSystems

ED11S

Expert Traditional Web Developer O11 (ES)

EDITION

Demo

QUESTIONS

9

FORMAT

Limited Edition PDF

Before you begin

What this is

9 practice questions for OutSystems **ED11S**, 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, **ED11S**
- the question number
- the email address on your order

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

QUESTION 1 SINGLE CHOICE

A developer needs to apply consistent theming to a Traditional Web application in OutSystems 11. Which framework provides the recommended UI components and styling for Traditional Web apps?

- ☐ A OutSystems UI (Silk UI)
- ☒ B OutSystems UI Web
- ☐ C Reactive Web Components
- ☐ D Material UI for OutSystems

ANSWER B

WHY For Traditional Web applications in OutSystems 11, the recommended and supported UI framework is 'OutSystems UI Web', the evolution of the former Silk UI for Traditional Web. 'OutSystems UI (Silk UI)' is typically associated with Reactive or older patterns, and 'Reactive Web Components' is the framework for Reactive Web apps. Material UI for OutSystems is not the official framework.

QUESTION 2 SINGLE CHOICE

In a Traditional Web screen's Preparation, what is the primary purpose of the Preparation action?

- ☐ A To define client-side JavaScript validations for input fields
- ☒ B To run server-side logic and aggregate data before the screen renders
- ☐ C To handle the screen's destination flow after submission
- ☐ D To define CSS classes used by the screen's widgets

ANSWER B

WHY The Preparation action of a Traditional Web screen runs on the server before the screen is rendered. It is the appropriate place to fetch data from the database, call server actions, and aggregate or compute values that the screen will display. Client-side validations belong in the screen's Validation, flow control belongs on the Destination, and CSS is handled through themes and stylesheets.

QUESTION 3

SINGLE CHOICE

Which Traditional Web screen callback is executed server-side when end-user input must be validated before the screen's main action runs?

- ☐ A Preparation
- ☒ B Validation
- ☐ C Destination
- ☐ D On Initialize

ANSWER B

WHY The Validation callback of a Traditional Web screen is the server-side callback that runs to validate user input before the screen's main action is executed. Preparation runs first to gather data, Validation runs after the screen submits but before the main action, and Destination controls where the user navigates after the action completes.

QUESTION 4

TRUE FALSE

In a Traditional Web application, every screen has a server-side Preparation, a server-side Validation, and a server-side Destination, and these run in that order when the user submits the screen.

- ☒ A True
- ☐ B False

ANSWER A

WHY True. The Traditional Web screen lifecycle executes the Preparation, then the Validation, and finally the Destination (which controls where the user navigates next) when the screen is submitted. This order is the documented lifecycle of Traditional Web screens.

QUESTION 5 SINGLE CHOICE

Which Traditional Web input widget allows the end user to pick a single value from a list of options rendered as a dropdown?

- ☐ A Combo Box
- ☐ B List Box
- ☐ C Checkbox Group
- ☒ D Radio Group

ANSWER D

WHY The Radio Group widget in Traditional Web (and in OutSystems UI Web) renders a set of mutually exclusive options, typically shown as a select/dropdown control bound to a static or dynamic list. A Combo Box is used for autocomplete-style single input, a List Box for multi-select lists, and a Checkbox Group for selecting multiple values. For a single-value dropdown, the Radio Group is the documented choice.

QUESTION 6 SINGLE CHOICE

In a Traditional Web screen, where should JavaScript code be placed so that it runs in the browser when the screen loads?

- ☐ A In the screen's Preparation action
- ☒ B In the JavaScript property of the screen or in an Expression of type JavaScript
- ☐ C In the module's Server Action
- ☐ D In the Database Entity's Default Value

ANSWER B

WHY Client-side JavaScript in Traditional Web is added via the screen's JavaScript property or via Expressions of type 'JavaScript' placed on widgets. The Preparation action runs on the server. Server Actions also run on the server, and entity default values are for database column defaults — none of these execute JavaScript in the browser.

QUESTION 7

SINGLE CHOICE

When defining navigation between two Traditional Web screens, which screen property must be set so that the destination screen's Destination flow is invoked after the source screen's main action runs?

- ☐ A Entry Point
- ☒ B Destination
- ☐ C Screen Reference
- ☐ D Allowed Roles

ANSWER B

WHY The 'Destination' property of a Traditional Web screen defines the flow that runs after the screen's main action completes; this is how the navigation chain to subsequent screens is configured. Entry Point marks the first screen of a flow, Screen Reference is used to embed content, and Allowed Roles controls access — none of these control post-action navigation.

QUESTION 8

SINGLE CHOICE

A Traditional Web developer needs to translate static text shown on a screen into multiple languages. Which mechanism should be used so the text is translatable in Service Center?

- ☐ A Hard-coded strings directly in the screen's labels
- ☐ B Locale-specific CSS files
- ☒ C Resources defined under the module's UI Flows / eSpace with translations configured for each locale
- ☐ D Storing translations in a database table and reading them in the Preparation

ANSWER C

WHY The supported, built-in way to translate UI text in OutSystems Traditional Web is through module Resources. Translations are managed per locale and surfaced automatically based on the user's effective locale. Hard-coded labels are not translatable, CSS files are styling, and while a database table is possible as a workaround, it is not the recommended pattern and does not integrate with the translation tooling.

QUESTION 9

SINGLE CHOICE

A Traditional Web developer needs to display a read-only field that combines a user's first name and last name on a screen. Which Expression output pattern correctly references these two attributes from the Preparation?

- ☒ **A** `GetUser.Current.FirstName + " " + GetUser.Current.LastName`
- ☐ **B** `FirstName + " " + LastName`
- ☐ **C** `User.FirstName + " " + User.LastName`
- ☐ **D** `GetUser.FirstName + " " + GetUser.LastName`

ANSWER A

WHY In OutSystems 11 Traditional Web, the Preparation aggregates a query (e.g., `GetUser`) into a List or single record. To reference an attribute on the screen you must navigate through the aggregate/action result (`GetUser.Current` for the current row), so option A is the only syntactically valid path.