

## **- Question Bank for Mobile Applications And Web APIs**

### **Elsewedy IATS**

#### **• Mobile Applications**

##### **1- True or False**

<b>True or False</b>		
#	<b>Question</b>	<b>Model Answer</b>
1	Cross-platform development frameworks allow developers to write code once and deploy it on multiple platforms, such as iOS and Android.	True
2	Flutter's hot reload feature requires a complete app restart.	False
3	Flutter can be used to develop applications for both Android and iOS.	True
4	The <code>build()</code> method in StateFull widget is called only once when the app starts.	False
5	The <code>Column</code> widget arranges its children in a horizontal layout.	False
6	The <code>pubspec.yaml</code> file in a Flutter project is used to manage dependencies.	True
7		
8	<code>GridView.builder</code> in Flutter that allows creating a grid with a fixed number of items.	False
9	To access data from a provider, you must always use the <code>Consumer</code> widget only in Flutter.	False
10	Using <code>Provider.of&lt;T&gt;(context, listen: false)</code> will rebuild the UI whenever the provider's state changes.	False
20	IntelliJ allows you to push your commits to a remote repository using the <b>Push</b> button or <code>git push</code> in the terminal.	True
41	<b>The <code>NetworkImage</code> class can be used directly in an <code>Image</code> widget to load images from the internet.</b>	True
42	<b>The <code>AssetImage</code> widget is used to load images from the internet.</b>	False
43	<b>A <code>TextField</code> in Flutter can only be used with a <code>TextEditingController</code>.</b>	False
44	<b>The <code>validator</code> property in a <code>TextField</code> is used to provide custom validation logic.</b>	True
45	<b>A <code>FloatingActionButton</code> must always have an <code>onPressed</code> callback to function correctly.</b>	True
46	<b>The <code>child</code> property in <code>FloatingActionButton</code> is used for the button to display content.</b>	True
47	<b>Using <code>async</code> and <code>await</code> in Dart allows asynchronous code to be written in a way that looks synchronous.</b>	True

48	<b>The Dio package supports both GET and POST HTTP methods for API requests.</b>	True
49	<b>If you do not use await with a Dio GET request, the response will still return synchronously.</b>	False
50	<b>The TextEditingController in a TextField is used to control and retrieve the entered text.</b>	True

## 2- Choose the correct answer

Choose the correct answer		
	Question	Model Answer
51	<p>The widget that represents a circular button for triggering primary actions in an application.</p> <ul style="list-style-type: none"> <li>a) Elevated Button</li> <li>b) text button</li> <li>c) Floating Action Button</li> <li>d) Material Button</li> </ul>	Floating Action Button
52	<p>The widget that Presents a scrollable list of widgets, ideal for displaying lists of items.</p> <ul style="list-style-type: none"> <li>a) ListView</li> <li>b) Container</li> <li>c) Column</li> <li>d) StateFull</li> </ul>	ListView
53	<p><b>What is the purpose of the build() method in a Flutter widget?</b></p> <ul style="list-style-type: none"> <li>a) To initialize variables</li> <li>b) To handle API calls</li> <li>c) To describe the widget's UI structure</li> <li>d) To navigate between screens</li> </ul>	c) To describe the widget's UI structure
54	<p><b>Which widget is used to manage the navigation and routing in a Flutter app?</b></p> <ul style="list-style-type: none"> <li>a) Column</li> <li>b) Navigator</li> <li>c) Drawer</li> <li>d) TabBar</li> </ul>	b) Navigator
55	<p><b>Which widget is typically used to arrange widgets vertically in Flutter?</b></p> <ul style="list-style-type: none"> <li>a) Row</li> <li>b) Stack</li> </ul>	c) Column

		c) Column d) Expanded	
56	What does Provider.of<T>(context, listen: true) do?  a) Accesses the provider's value without rebuilding b) Accesses and listens for changes in the provider's value c) Removes the provider from memory d) Deletes the current widget	b) Accesses and listens for changes in the provider's value	
57	What is the return type when parsing a JSON object in Dart?  a) List b) String c) Map d) int	c) Map	
58	<b>Which widget is used as the root of a Flutter application?</b>  a) StatelessWidget b) StatefulWidget c) MaterialApp d) Scaffold	c) MaterialApp	
61	<b>What is the main purpose of the consumer widget in Flutter when using Provider for state management?</b>  A) It listens to changes in the state and rebuilds the widget when the state changes. B) It updates the state directly when the user interacts with the widget. C) It allows access to state without rebuilding the widget. D) It manages the state and notifies the listeners automatically.	A) It listens to changes in the state and rebuilds the widget when the state changes	
62	<b>In Flutter, the Row widget aligns its children along the ..... axis by default.</b>  A) Vertical B) Horizontal C) Main D) Cross	B) Horizontal	
73	<b>Which widget is used to display an image from the internet in Flutter?</b>  a) AssetImage b) Image.network c) Imageassets d) Image.asset	b) Image.network	
74	<b>What is the correct way to load an image from the assets folder?</b>  a) NetworkImage('assets/image.png') b) Image.network('assets/image.png')	c) AssetImage('assets/image.png') )	

		c) AssetImage('assets/image.png')  d) network. Image.('assets/image.png')	
	75	<b>Which property of a TextField widget is used to validate user input?</b> a) onChanged b) decoration c) controller d) validator	d) validator
	76	<b>What does the FloatingActionButton widget primarily provide in a Flutter app?</b> a) A persistent drawer b) A scrollable list c) A button for primary actions d) An area for app notifications	c) A button for primary action
	77	<b>Which parameter is required for the FloatingActionButton widget?</b> a) tooltip b) child c) elevation d) onPressed	d) onPressed
	78	<b>What is the purpose of the async keyword in Dart?</b> a) To call the function b) To make the function asynchronous c) To declare a constant function d) To handle errors automatically	b) To make the function asynchronous
	79	<b>Which package is commonly used in Flutter for making HTTP GET and POST requests?</b> a) provider b) firebase_auth c) dio d) connectivity	c) dio
	80	<b>How do you handle an asynchronous GET request using the Dio package?</b> a) dio.get() without await b) Use await dio.get() inside an async function c) Use dio.post() instead of get() d) Only use synchronous functions	b) Use await dio.get() inside an async function
	81	<b>What is the purpose of the validator property in a TextField widget?</b> a) To format the text inside the field b) To verify the input and return an error message if invalid c) To manage focus changes d) To control the text entry behavior	b) To verify the input and return an error message if invalid

Complete in gaps with a word from the following words between brackets

(Consumer – Provider.of - rebuild automatically – rebuild – GridView - ListView.builder - Image.asset - floating action button -Title - actions - leading - Async - await - MultiProvider – data – itemBuilder -JSON response - Data model - Service – Provider - Design analysis – widgets -screens – postman - ChangeNotifierProvider )

	<b>Question</b>	<b>Model Answer</b>
82	To access the state using Provider, you can use either ..... or ..... to listen for changes.	Consumer or Provider.of
83	In Provider, setting listen to true means the widget will ..... when the state changes, while listen set to false means the widget will not .....	rebuild automatically, rebuild
84	To create a grid layout in Flutter, you would typically use the widget ..... for a flexible and scrollable layout.	GridView
85	A ListView in Flutter can be used to display a scrollable list of items, and it can be optimized by using ..... for lazy loading.	ListView.builder
86	In Flutter, to display an image from the local assets, you use the ..... widget.	Image.asset
87	The Scaffold widget in Flutter provides a basic visual layout structure for the app and includes features like an app bar, body, and .....	floating action button
88	The AppBar widget can be customized with various properties like ..... , ..... and ..... for adding widgets to the left side of the app bar.	Title , actions , leading
89	In Dart, to handle asynchronous code execution, you typically use the keywords ..... and .....	Async , await
90	To handle multiple providers in a large Flutter app, you use the ..... widget at the root of the widget tree to organize them.	MultiProvider
91	To send data using a POST request with Dio, you should use the method dio.post and pass the data through the ..... parameter.	data
92	When using a ListView.builder, the ..... callback is used to generate items for the list.	itemBuilder
93	When building a flutter app the first step is to check ..... returned from the api	JSON response

94	When building a flutter app the second step is to create an appropriate ..... for the returned data from the api.	Data model
95	When building a flutter app the third step is to create a ..... file to send requests and receive responses from the api.	Service
96	When building a flutter app the forth step is to create a ..... file to handle the model and services in the UI	Provider
98	When building a flutter app the sixth step is to create a ..... files to create the widget item that is repeated in the design.	widgets
100	We use ..... to test the api by sending requests and getting responses.	postman
101	To create and use a provider object we wrap the material app with .....	ChangeNotifierProvider

..

Question	Answer
1- What is the purpose of the build() method in Flutter?	The build() method returns the UI of the widget.
2- What is the difference between const and final in Flutter?	final is set once at runtime, while const is set at compile time and never changes.
3- Why do we use Provider in Flutter instead of passing data between widgets manually?	Provider is used to share data between widgets easily without passing it manually.
4- What is the difference between listen: true and listen: false in Provider.of()?	listen: true rebuilds the widget when data changes, while listen: false does not.
5- How do you fetch data from an API in Flutter using Provider? Explain the steps.	Fetch the data in a Provider class, save it, call notifyListeners(), then use it in the UI.
6- Why do we usually fetch API data inside a Provider class and not directly inside the UI widget?	To keep logic separate from UI and make the code cleaner.
7- What is the difference between push(), pop(), and pushReplacement()?	<ul style="list-style-type: none"> <li>• <b>push()</b> Adds a new screen to the navigation stack and moves to it. The previous screen remains in memory.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>pop()</b> Removes the current screen from the stack and returns to the previous screen.</li>   <li>• <b>pushReplacement()</b> Replaces the current screen with a new one. The user cannot return to the previous screen using the back button.</li> </ul>
8) What is the purpose of the SafeArea widget?	<p>The SafeArea widget prevents UI elements from being hidden behind system areas such as:</p> <ul style="list-style-type: none"> <li>• the notch</li> <li>• status bar</li> <li>• navigation bar</li> </ul> <p>It automatically adds padding so the content appears in the visible and safe part of the screen.</p>
9) Explain the difference between Padding and Margin in Flutter.	<ul style="list-style-type: none"> <li>• <b>Padding</b> Adds space inside a widget, between the widget's content and its border.</li> <li>• <b>Margin</b> Adds space outside a widget, between the widget and surrounding widgets. (In Flutter, margin is applied using the Container widget.)</li> </ul>
10) How do you pass data between screens in Flutter?	<p>Data can be passed between screens by:</p> <ul style="list-style-type: none"> <li>• Passing values through the constructor when navigating using Navigator.push()</li> <li>• Using named routes with arguments</li> <li>• Using state management solutions (like Provider) for shared data</li> </ul>
11) Why do we use ListView.builder instead of ListView in Flutter?	<p><b>ListView.builder:</b></p> <ul style="list-style-type: none"> <li>• creates list items only when they are visible on the screen</li> <li>• improves performance</li> <li>• uses less memory</li> </ul> <p>It is recommended for large or dynamic lists, while ListView is suitable for small, static</p>
13. Explain the difference between StatefulWidget and StatelessWidget	<p><b>13. StatelessWidget vs StatefulWidget</b></p> <p><b> StatelessWidget:</b> UI does not change after build.</p> <p><b> StatefulWidget:</b> UI can change using setState().</p>

<b>14. Explain ChangeNotifier and notifyListeners() in Flutter</b>	<p><b>14. ChangeNotifier and notifyListeners()</b></p> <p><b>ChangeNotifier:</b> Used for state management.</p> <p><b>notifyListeners():</b> Updates listening widgets when data changes.</p>
<b>15. Explain the difference between Row and Column widgets in Flutter</b>	<p><b>15. Row vs Column</b></p> <p><b>Row:</b> Arranges widgets horizontally.</p> <p><b>Column:</b> Arranges widgets vertically.</p>
<b>16. Explain the Scaffold widget in Flutter</b>	<p><b>16. Scaffold widget</b></p> <p>Provides the basic screen layout for a Flutter app.</p>
<b>17. Explain the main components of the Scaffold widget</b>	<p><b>17. Main components of Scaffold</b></p> <p><b>AppBar</b></p> <p><b>Body</b></p> <p><b>FloatingActionButton</b></p> <p><b>Drawer</b></p> <p><b>BottomNavigationBar</b></p>
<b>18. What is the role of GestureDetector widget?</b>	<p><b>18. GestureDetector widget</b></p> <p>Detects user gestures like tap, swipe, and long press.</p>
<b>19. What is Scaffold in Flutter? What is its purpose?</b>	<p><b>19-Scaffold is a layout widget that provides the basic visual structure of a Flutter screen.</b></p> <p><b>Purpose:</b></p> <p>Holds common UI elements such as:</p> <p><b>AppBar</b></p> <p><b>Drawer</b></p> <p><b>BottomNavigationBar</b></p> <p><b>FloatingActionButton</b></p> <p>Ensures proper layout and material design behavior.</p> <p>Example use: Creating a standard page layout.</p>
<b>20. What is BuildContext? Why is it important in</b>	<b>20- BuildContext is a reference to the location of a</b>

<b>Flutter?</b>	<p><b>widget in the widget tree.</b></p> <p><b>Importance:</b></p> <p>Allows access to:</p> <ul style="list-style-type: none"> <li>Theme data</li> <li>MediaQuery (screen size)</li> <li>Navigator (for navigation)</li> <li>Inherited widgets</li> </ul> <p>Used to build widgets and interact with the widget tree. Without BuildContext, Flutter cannot determine where the widget belongs.</p>
<b>21.What is async / await in Flutter? Why is it important?</b>	<p><b>21- async and await are used for asynchronous programming.</b></p> <p><b>Purpose:</b></p> <p>Handle time-consuming operations such as:</p> <ul style="list-style-type: none"> <li>API calls</li> <li>File operations</li> <li>Database queries</li> <li>Prevent UI freezing.</li> </ul> <p><b>Importance:</b></p> <p>Keeps the app responsive Makes asynchronous code readable and easier to manage</p>
<b>22. Explain the purpose of the setState() method.</b> <b>setState() is used in StatefulWidget to notify Flutter that the UI needs to be rebuilt.</b>	<p><b>22- Purpose:</b></p> <p>Updates the UI when data changes Triggers the build() method again Used when: Changing values like counters, form inputs, toggles, etc.</p>

..

## ASP.NET Web API Question Bank (100 Questions)

---

### Section 1: True/False Questions (25)

#	Question	Answer
1	Asynchronous operations improve scalability in Web API.	True
2	RESTful APIs require maintaining client sessions on the server.	False
3	Singleton lifetime provides a single instance used throughout the application.	True
4	Stateless APIs store user data across requests.	False
5	Lazy loading loads related data only when explicitly accessed.	True
6	Dependency Injection simplifies testing and enhances code maintainability.	True
7	HTTPS ensures both data encryption and integrity.	True
8	Middleware components execute in the order they are added in the pipeline.	True
9	Indexing always improves performance without any trade-offs.	False
10	Tokens in stateless authentication are stored on the server.	False
11	Data annotations can be used to enforce validation rules on models.	True
12	Transient services are created once and reused for all requests.	False
13	Explicit loading requires an additional call to load related data.	True
14	Middleware can be used for logging, authentication, and error handling.	True
15	Using DTOs prevents exposing internal domain models directly.	True
16	PUT method is used to partially update an existing resource.	False
17	HTTPS is optional for securing data in production environments.	False
18	Dependency Injection helps in reducing code coupling.	True
19	Eager loading may reduce the number of database calls.	True
20	JWT tokens are stored in the Web API server memory.	False
21	Middleware sequence affects how requests and responses are handled.	True
22	The PATCH method is used to fully replace a resource.	False
23	CORS issues occur when making cross-origin requests without proper headers.	True
24	Data seeding helps initialize a database with default data.	True
25	Scoped lifetime creates a new instance for each service call within the same request.	True

---

### Section 2: Multiple Choice Questions (30)

#	Question	Choices	Answer
26	Which data loading strategy loads related data when requested?	a) Eager b) Lazy c) Explicit d) None	c

<b>27</b>	Which HTTP method is used to update an existing resource completely?	a) GET b) POST c) PUT d) PATCH	c
<b>28</b>	What is the main purpose of the Repository Pattern?	a) Manage UI b) Abstract data access c) Handle routing	b
<b>29</b>	Which middleware handles request routing?	a) UseRouting b) UseCors c) UseAuthorization	a
<b>30</b>	Which DI lifetime provides a new instance for each request?	a) Singleton b) Scoped c) Transient	b
<b>31</b>	Which annotation ensures a property is mandatory?	a) [Range] b) [Required] c) [MaxLength]	b
<b>32</b>	Which status code indicates "Unauthorized" access?	a) 200 b) 401 c) 404	b
<b>33</b>	Which token type is commonly used in stateless Web API authentication?	a) OAuth b) JWT c) Session	b
<b>34</b>	Which is NOT a valid data loading strategy?	a) Eager b) Lazy c) Implicit d) Explicit	c
<b>35</b>	Middleware can:	a) Handle requests b) Process responses c) Both	c
<b>36</b>	Which is NOT a valid parameter binding source?	a) FromBody b) FromQuery c) FromRoute d) FromLocal	d
<b>37</b>	Which attribute limits the maximum length of a string?	a) [Required] b) [MaxLength] c) [Range]	b
<b>38</b>	Fluent API is primarily used to:	a) Configure models b) Handle logging	a
<b>39</b>	Which HTTP status code represents resource creation?	a) 200 b) 201 c) 400	b
<b>40</b>	DTOs help to:	a) Expose domain logic b) Limit data exposure	b
<b>41</b>	Which method supports eager loading?	a) Include b) Query	a
<b>42</b>	Which HTTP method retrieves data without modifying it?	a) GET b) PUT c) DELETE	a
<b>43</b>	Which command generates migration files?	a) add-migration b) update-database	a
<b>44</b>	Which annotation validates a range of values?	a) [Range] b) [MaxLength]	a
<b>45</b>	How can HTTPS be enforced in a Web API project?	a) UseHttpsRedirection b) EnableSSL	a
<b>46</b>	Which lifetime creates a single instance for the entire application?	a) Singleton b) Scoped c) Transient	a
<b>47</b>	Which method partially updates a resource?	a) PUT b) PATCH c) DELETE	b
<b>48</b>	Which HTTP status code indicates a server error?	a) 500 b) 404 c) 403	a
<b>49</b>	What is the benefit of using data annotations?	a) Styling b) Validation c) Performance	b

50	Which port does HTTPS commonly use?	a) 80 b) 443 c) 8080	b
51	Which annotation restricts the range of integer inputs?	a) [MaxLength] b) [Range] c) [Required]	b
52	Which is the correct command to apply migrations to a database?	a) apply-db b) update-database c) run-migrate	b
53	What is the primary use of CORS in Web API?	a) Logging b) Security c) Cross-origin requests	c
54	Which of the following handles JSON serialization in ASP.NET Web API?	a) Newtonsoft.Json b) System.IO c) JsonWriter	a
55	Which status code indicates "Forbidden"?	a) 403 b) 404 c) 500	a

---

### Section 3: Matching Questions – Part 1 (15) (*Column B randomized, then alphabetically ordered with correct answers*)

#	Column A	Column B	Answer
56	Eager Loading	A. Ensures a value is provided	H
57	Lazy Loading	B. Loads data only when accessed	B
58	Explicit Loading	C. Indicates resource not found	F
59	Singleton Lifetime	D. New instance per request	I
60	Scoped Lifetime	E. Limits string length to 50 characters	D
61	Transient Lifetime	F. Requires manual loading call	G
62	[Required]	G. New instance every time requested	A
63	[MaxLength(50)]	H. Loads related data immediately	E
64	404 Not Found	I. Single instance throughout application	C

65	500 Internal Server Error	J. Indicates server-side failure	J
66	UseCors	K. Allows cross-origin requests	K
67	HTTPS	L. Provides stateless authentication	O
68	Middleware Pipeline	M. Validates incoming model data	N
69	ModelState	N. Processes requests and responses	M
70	JWT	O. Secures communication	L

---

## Section 4: Matching Questions – Part 2 (15)

#	Column A	Column B	Answer
71	Postman	A. Fully replaces a resource	B
72	DTO	B. API testing tool	F
73	UseRouting	C. Validates numeric value ranges	I
74	[Range]	D. Redirects HTTP to HTTPS	C
75	PATCH Method	E. Manages resource access	H
76	PUT Method	F. Transfers limited data between layers	A
77	HTTPS Redirection	G. Carries claims-based authentication	D
78	UseAuthorization	H. Updates parts of a resource	E
79	JWT Token		G

		I. Enables request routing	
80	CORS	J. Handles cross-origin policies	J
81	Include Method	K. Loads related entities eagerly	K
82	FromBody Attribute	L. Maps request to endpoint handlers	M
83	UseEndpoints	M. Binds data from request body	L
84	UseExceptionHandler	N. Restricts string property length	O
85	[StringLength]	O. Catches unhandled exceptions	N

## Section 5: Steps, Concepts & Scenarios (15) (*Detailed answers with step-by-step explanations*)

#	Question	Detailed Answer
86	List the stages of a Web API request lifecycle.	1. Routing 2. Controller Selection 3. Action Method Execution 4. Data Processing Through Repository 5. Response
87	Mention categories of HTTP status codes.	- 1xx: Informational - 2xx: Success - 3xx: Redirection - 4xx: Client Errors - 5xx: Server Errors
88	How is Fluent API used in model configurations?	Fluent API is used within the OnModelCreating method to configure relationships, keys, and constraints programmatically instead of using attributes.
89	Name Dependency Injection lifetimes.	- Singleton: Single instance throughout application - Scoped: New instance per request - Transient: New instance every service call
90	Describe data loading strategies.	- Eager Loading: Loads related data with the main query - Lazy Loading: Loads related data when accessed - Explicit Loading: Loads related data on-demand via commands

<b>91</b>	How does middleware order affect processing?	Middleware runs sequentially; earlier middleware can block or modify requests before they reach later components.
<b>92</b>	How to implement JWT authentication?	Steps: 1. Add authentication services to DI container 2. Configure JWT parameters (issuer, audience, key) 3. Use authentication and authorization middleware 4. Generate tokens using claims.
<b>93</b>	How can you validate incoming requests?	Use data annotations like [Required], [Range] or check ModelState.IsValid in controllers.
<b>94</b>	Steps to enable HTTPS redirection.	1. Add app.UseHttpsRedirection() in Configure method 2. Ensure SSL certificate is configured.
<b>95</b>	How are migration changes applied?	Use commands: - add-migration <name> to create migration - update-database to apply changes to DB.
<b>96</b>	Difference between PUT and PATCH methods.	- PUT: Full resource update - PATCH: Partial resource update.
<b>97</b>	Why is eager loading beneficial?	Reduces multiple round trips to the database by loading related data in one query.
<b>98</b>	What is the use of CORS?	CORS enables controlled access to resources from different origins, improving security while allowing flexibility.
<b>99</b>	How does indexing affect database performance?	Indexing speeds up read operations but can slow down write operations and consumes storage.
<b>100</b>	Why use DTOs in Web API?	DTOs prevent exposing internal models, reduce data sent over the network, and improve security and maintainability.