

## AppDev2

Section Id :	64065338411
Section Number :	14
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	33
Number of Questions to be attempted :	33
Section Marks :	100
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065381004
Question Shuffling Allowed :	No
Is Section Default? :	null

**Question Number : 379 Question Id : 640653566365 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 0**

Question Label : Multiple Choice Question

**THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MODERN APPLICATION DEVELOPMENT 2 (COMPUTER BASED EXAM)"**

**ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?**

**CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.**

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE [TOP](#) FOR THE SUBJECTS REGISTERED BY YOU)

**Options :**

6406531892825. ✔ YES

6406531892826. ✖ NO

<b>Sub-Section Number :</b>	2
<b>Sub-Section Id :</b>	64065381005
<b>Question Shuffling Allowed :</b>	Yes
<b>Is Section Default? :</b>	null

**Question Number : 380 Question Id : 640653566366 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are false regarding javascript language?

**Options :**

6406531892827. ✔ Hoisting moves the declaration of arrow functions to the top of their scopes.

6406531892828. ✖ All 3 keywords, i.e., "var", "let" and "const" can be used to declare global variables.

6406531892829. ✖ The functions are treated as first class citizens in the language.

6406531892830. ✔ Node.js is an example of a javascript engine.

**Question Number : 381 Question Id : 640653566377 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are true about cookies and CORS?

**Options :**

6406531892871. ✓ A session cookie expires as soon as the browser is closed.

6406531892872. ✓ The CORS mechanism reduces the chances of malicious actions by explicitly saying which URLs can be the originators of data.

6406531892873. ✗ A browser sandbox causes any browser based malware to directly affect the user's system.

6406531892874. ✗ All of these

**Question Number : 382 Question Id : 640653566384 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2 Selectable Option : 0**

Question Label : Multiple Select Question

Suppose a script is downloaded from the origin '<http://example.com>'. The script can not make a request to which of the following origins, assuming none of the origins allow cross-origin requests?

**Options :**

6406531892897. ✗ <http://example.com/profile>

6406531892898. ✓ <http://new.example.com>

6406531892899. ✗ <http://example.com>

6406531892900. ✓ <http://new.example.com/profile>

**Question Number : 383 Question Id : 640653566385 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction**

**Time : 0**

**Correct Marks : 2 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are false about pub/sub messaging?

**Options :**

6406531892901. ✔ Publisher has to know about all the subscribers.

6406531892902. ✖ Publisher does not have to know about the subscribers.

6406531892903. ✖ Communication between publisher and subscribers is asynchronous.

6406531892904. ✔ Communication between publishers and subscribers is synchronous.

**Question Number : 384 Question Id : 640653566386 Question Type : MSQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction**

**Time : 0**

**Correct Marks : 2 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are false?

**Options :**

6406531892905. ✔ In short polling, the server does not respond until the data is available or the request times out.

6406531892906. ✖ In long polling, the server does not respond until the data is available or the request times out.

6406531892907. ✖ In short polling, the server responds immediately with or without data.

6406531892908. ✔ In long polling, the server responds immediately with or without data.

**Sub-Section Number :** 3

**Sub-Section Id :** 64065381006

**Question Shuffling Allowed :** Yes

**Is Section Default? :** null

**Question Number : 385 Question Id : 640653566367 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Which of the following shows the correct output if the javascript program written below is executed?

```
const a = [2, 5, 8, 11, 14, 17]
const b = a.map(element => element ** 2)
const c = b.filter(element => element % 2)
const d = c.find(element => element % 3 == 1)
console.log(d)
```

**Options :**

6406531892831. ✖ [25, 121, 289]

6406531892832. ✖ [25, 121]

6406531892833. ✔ 25

6406531892834. ✖ 25 121

**Question Number : 386 Question Id : 640653566374 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Which of the following statements is true in the context of REST?

**Options :**

6406531892859. ✖ Both the HTTP request methods, "GET" and "POST" are idempotent.

6406531892860. ✖ In general, an HTTP "PUT" request is used to make incremental changes to an existing resource.

6406531892861. ✓ A "DELETE" request, that deletes a resource using a unique identifier, is idempotent.

6406531892862. ✖ None of these

**Question Number : 387 Question Id : 640653566379 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Which of the following shows the correct output if the javascript program written below is executed?

```
async function demo () {  
    let prom = new Promise((res, rej) => {  
        setTimeout(() => res(40), 2000)  
    })  
    console.log("Statement 2") // Statement 2  
    return prom  
}  
  
demo().then(result => console.log(result))  
console.log("Statement 1") // Statement 1
```

**Options :**

40  
Statement 2  
6406531892879. ✖ Statement 1

40  
Statement 1  
6406531892880. ✖ Statement 2

Statement 1  
Statement 2  
6406531892881. ✖ 40

Statement 2

Statement 1

6406531892882. ✓ 40

**Question Number : 388 Question Id : 640653566383 Question Type : MCQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Which of the following statements is false regarding a message broker?

**Options :**

6406531892893. ✓ A message broker allows two servers in a network to directly communicate with each other, without an intermediary.

6406531892894. ✗ A message broker makes the network scalable for adding more servers to the network.

6406531892895. ✗ A message broker can be used for batch processing of messages.

6406531892896. ✗ A message broker is well suited in case of traffic spikes, as messages are retained in the queue until processed.

**Question Number : 389 Question Id : 640653566387 Question Type : MCQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Which of the following statement(s) is incorrect regarding the prototype in JavaScript?

**Options :**

6406531892909. ✗ Every constructor function has a property named 'prototype'.

6406531892910. ✖ Any object created using the new keyword and the constructor function, will inherit from the constructor's prototype object.

6406531892911. ✖ Prototype of an object can be accessed using the '\_\_proto\_\_' property of the object.

6406531892912. ✔ The prototype of an object can be accessed using the 'prototype' property of the object.

**Question Number : 390 Question Id : 640653566390 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup "index.html" and javascript file "app.js".



index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
const Player = {
  template: `<div><div>{{striker.name}}*</div>
<div>{{nonStriker.name}}</div>
</div>`,
  data() {
    return {
      player1: {name: 'Rohit'},
      player2: {name: 'Virat'},
    }
  },
  props: {
    run: { type: Number, default: 0 },
  },
  computed: {
    striker() {
      return this.run % 2 == 0 ? this.player1 : this.player2
    },
    nonStriker() {
      return this.run % 2 == 1 ? this.player1 : this.player2
    },
  },
}

const Run = {
  template: `<button @click="$emit('run-changed',
run)">{{run}}</button>`,
  props: { run: Number },
}

new Vue({
  el: '#app',
  template: `<div>
    <Player :run='score' />
    <div style='display: flex'>
      <Run :run="3" @run-changed = 'runChanged' />
      <Run :run="4" @run-changed = 'runChanged' />
    </div>
  </div>`,
  components: { Run, Player },
  data: {
    score: 0,
  },
  methods: {
    runChanged(run) {
      this.score = run
    },
  },
})
```

Suppose the user clicks on buttons with text “3” and then “4”, what will be rendered inside the “Player” component?

**Options :**

Rohit\*

6406531892921. ✓ Virat

Virat\*  
6406531892922. ✖ Rohit

Virat\*  
6406531892923. ✖ Rohit\*

Rohit\*  
6406531892924. ✖ Virat\*

**Question Number : 391 Question Id : 640653566391 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup "index.html" and javascript file "app.js".

index.html:

```
<div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3.0.0/dist/vue-router.js"></script>
  <script src="app.js" type="module"></script>
```

app.js:

```
const data = {
  1: { totalRun: 1000, totalMatches: 20 },
  2: { totalRun: 7000, totalMatches: 100 },
}
const NotFound = { template: `<div> Player Not Found</div>` }
const Profile = {
  template: `<div>Run: {{stat.totalRun}}, Matches:
{{stat.totalMatches}}, Average: {{average}}</div>`,
  data() {
    return {
      stat: data[this.$route.params.id],
    }
  },
  computed: {
    average() {
      return this.stat.totalRun / this.stat.totalMatches
    },
  },
}
const router = new VueRouter({
  routes: [
    { path: '/profile/:id', component: Profile },
    { path: '*', component: NotFound },
  ],
})

new Vue({
  el: '#app',
  template: '<div><router-view /></div>',
  router,
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be rendered inside the "router-view" component for the URL "<http://127.0.0.1:8080/#/player>"?

**Options :**

6406531892925. ✔ Player Not Found

6406531892926. ✖ Run: 1000, Matches: 20, Average: 50

6406531892927. ✖ Run: 7000, Matches: 100, Average: 70

6406531892928. ✖ None of these

**Question Number : 392 Question Id : 640653566392 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup “index.html” and javascript file “app.js”.

index.html:

```
<div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3.0.0/dist/vue-router.js"></script>
  <script src="app.js" type="module"></script>
```

app.js:

```
const data = {
  1: { totalRun: 1000, totalMatches: 20 },
  2: { totalRun: 7000, totalMatches: 100 },
}
const NotFound = { template: `<div> Player Not Found</div>` }
const Profile = {
  template: `<div>Run: {{stat.totalRun}}, Matches:
{{stat.totalMatches}}, Average: {{average}}</div>`,
  data() {
    return {
      stat: data[this.$route.params.id],
    },
  },
  computed: {
    average() {
      return this.stat.totalRun / this.stat.totalMatches
    },
  },
}
const router = new VueRouter({
  routes: [
    { path: '/profile/:id', component: Profile },
    { path: '*', component: NotFound },
  ],
})
new Vue({
  el: '#app',
  template: '<div><router-view /></div>',
  router,
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be displayed for the URL "<http://127.0.0.1:8080/#/profile/1>"?

**Options :**

6406531892929. ✖ Player Not Found

6406531892930. ✔ Run: 1000, Matches: 20, Average: 50

6406531892931. ✖ Run: 7000, Matches: 100, Average: 70

6406531892932. ✖ None of these

**Question Number : 393 Question Id : 640653566393 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup “index.html” and javascript file “app.js”.

index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3.0.0/dist/vue-router.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
const overs = {
  1: [2, 3, 4, 6, 2],
  2: [3, 2, 5, 3, 1],
}

const NotFound = { template: `<div> Over Not Found</div>` }
const MatchNotStarted = { template: `<div>Match has not yet started.
</div>` }

const currentOver = {
  template: `<div style='display: flex'>
<div v-for='run in over' style='padding: 10px'>{{run}}</div>
</div>`,
  data() {
    return {
      over: overs[this.$route.params.overNo]
        ? overs[this.$route.params.overNo]
        : overs[2],
    },
  },
}

const liveScore = {
  template: `<div>Total Score:{{totalScore}} <router-view /></div>`,
  data() {
    return {
      totalScore: 100,
    },
  },
}

const router = new VueRouter({
  routes: [
    {
      path: '/live-score',
      component: liveScore,
      children: [
        { path: '', component: MatchNotStarted },
        { path: 'over/:overNo', component: currentOver },
      ],
    },
  ],
})

new Vue({
  el: '#app',
  template: `<div><router-view /></div>`,
  router,
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be rendered in the "router-view" component of the "live-score" component for URL "http://localhost:8080/#/live-score"?

Options :

6406531892933. ✖ Total Score:100

6406531892934. ✔ Match has not yet started.

6406531892935. ✖ 2 3 4 6 2

6406531892936. ✖ Over Not Found

**Question Number : 394 Question Id : 640653566394 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup “index.html” and javascript file “app.js”.



index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3.0.0/dist/vue-router.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
const overs = {
  1: [2, 3, 4, 6, 2],
  2: [3, 2, 5, 3, 1],
}

const NotFound = { template: `<div> Over Not Found</div>` }
const MatchNotStarted = { template: `<div>Match has not yet started.
</div>` }

const currentOver = {
  template: `<div style='display: flex'>
<div v-for='run in over' style='padding: 10px'>{{run}}</div>
</div>`,
  data() {
    return {
      over: overs[this.$route.params.overNo]
        ? overs[this.$route.params.overNo]
        : overs[2],
    }
  },
}

const liveScore = {
  template: `<div>Total Score:{{totalScore}} <router-view /></div>`,
  data() {
    return {
      totalScore: 100,
    }
  },
}

const router = new VueRouter({
  routes: [
    {
      path: '/live-score',
      component: liveScore,
      children: [
        { path: '', component: MatchNotStarted },
        { path: 'over/:overNo', component: currentOver },
      ],
    },
  ],
})

new Vue({
  el: '#app',
  template: `<div><router-view /></div>`,
  router,
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be rendered in the "router-view" component of the "live-score" component for URL "http://localhost:8080/#/live-score/over/2"?

Options :

6406531892937. ✖ Total Score:100

6406531892938. ✖ Match has not yet started.

6406531892939. ✖ 2 3 4 6 2

6406531892940. ✔ 3 2 5 3 1

**Question Number : 395 Question Id : 640653566395 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the below application with markup “index.html” and javascript file “app.js”.

index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3.0.0/dist/vue-router.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
const overs = {
  1: [2, 3, 4, 6, 2],
  2: [3, 2, 5, 3, 1],
}

const NotFound = { template: `<div> Over Not Found</div>` }
const MatchNotStarted = { template: `<div>Match has not yet started.
</div>` }

const currentOver = {
  template: `<div style='display: flex'>
<div v-for='run in over' style='padding: 10px'>{{run}}</div>
</div>`,
  data() {
    return {
      over: overs[this.$route.params.overNo]
        ? overs[this.$route.params.overNo]
        : overs[2],
    }
  },
}

const liveScore = {
  template: `<div>Total Score:{{totalScore}} <router-view /></div>`,
  data() {
    return {
      totalScore: 100,
    }
  },
}

const router = new VueRouter({
  routes: [
    {
      path: '/live-score',
      component: liveScore,
      children: [
        { path: '', component: MatchNotStarted },
        { path: 'over/:overNo', component: currentOver },
      ],
    },
  ],
})

new Vue({
  el: '#app',
  template: `<div><router-view /></div>`,
  router,
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be rendered in the "router-view" component of the "live-score" component for URL "<http://localhost:8080/#/live-score/over/20>"?

Options :

6406531892941. ✖ Total Score:100

6406531892942. ✖ Match has not yet started.

6406531892943. ✖ 2 3 4 6 2

6406531892944. ✔ 3 2 5 3 1

Sub-Section Number :	4
Sub-Section Id :	64065381007
Question Shuffling Allowed :	Yes
Is Section Default? :	null

**Question Number : 396 Question Id : 640653566368 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following is/are not the correct ways to achieve the following.

- 1. Always apply class named "errorClass",
- 2. The class named "activeClass" should only be applied when the Vue data variable "isActive" is truthy

**Options :**

6406531892835. ✖ `<div :class="[isActive ? 'activeClass' : ' ', 'errorClass']"></div>`

6406531892836. ✖ `<div :class="{ 'activeClass' : isActive }, 'errorClass' "></div>`

6406531892837. ✔ `<div :class="[activeClass ? isActive : ' ', 'errorClass']"></div>`

6406531892838. ✔ `<div :class="{ isActive : 'activeClass' }, 'errorClass' "></div>`

**Question Number : 397 Question Id : 640653566373 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are false regarding Vuex?

**Options :**

6406531892855. ✖ A Vuex store provides a single source of truth that can drive the application.

6406531892856. ✔ It provides a variable named "this.\$vuexstore" to allow components to access the store data.

6406531892857. ✔ A component cannot have its own local state, if the application uses Vuex.

6406531892858. ✖ The mutations and actions are some constructs of a Vuex store.

**Question Number : 398 Question Id : 640653566376 Question Type : MSQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are false regarding forward and reverse proxy?

**Options :**

6406531892867. ✔ A forward proxy is used to protect the server(s) from the outside world.

6406531892868. ✖ A reverse proxy can be used to cache the responses, to reuse them for similar subsequent requests.

6406531892869. ✖ A load balancer becomes irrelevant, where there is not more than 1 server.

6406531892870. ✔ An efficient load balancer must use a round-robin algorithm.

**Question Number : 399 Question Id : 640653566380 Question Type : MSQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Selectable Option : 0**

Question Label : Multiple Select Question

Suppose you are building an application which requires sending emails to all the users stored in the database at regular intervals. Given below are the two approaches to satisfy the requirement.

Approach 1:

```
@celery.task
def send_bulk_emails():
    all_users = User.query.all() // Get all the user objects from the
    database
    for user in all_users:
        // send email to the user

send_bulk_emails.delay() // invokes celery task
```

Approach 2:

```
@celery.task
def send_email(email):
    // send email to the user

all_users = User.query.all() // Get all the user objects from the database
for user in all_users:
    send_email.delay(user.email) // invokes celery task
```

Choose the correct statement(s).

**Options :**

6406531892883. ✔ Approach 1 will take more time than approach 2, if more than 2 celery workers are available.

6406531892884. ✖ Approach 2 will take more time than approach 1, if more than 2 celery workers are available.

6406531892885. ✔ Both the approaches will take same time (approx), if only 1 celery worker is available.

6406531892886. ✖ None of these

**Question Number : 400 Question Id : 640653566381 Question Type : MSQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following statement(s) is/are correct regarding Vue.js framework?

**Options :**

6406531892887. ✔ In Vue 2, the “el” property of the Vue constructor can be used to refer to an HTML element via an ID or a class.

6406531892888. ✔ The components of Vue must define “data” property as a function.

6406531892889. ✖ A Vue component cannot implement lifecycle hooks (i.e., created, mounted, etc.), if they are already implemented in the Vue instance.

6406531892890. ✖ All of these

<b>Sub-Section Number :</b>	5
<b>Sub-Section Id :</b>	64065381008
<b>Question Shuffling Allowed :</b>	Yes
<b>Is Section Default? :</b>	null

**Question Number : 401 Question Id : 640653566369 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Which of the following statements is false about VueJS framework?

**Options :**

6406531892839. ✖ VueJS is a framework built on top of HTML, CSS and javascript.

6406531892840. ✖ VueJS is built on a component based architecture.

6406531892841. ✔ It cannot be installed/used via npm (node package manager).

6406531892842. ✖ All of these



**Question Number : 402 Question Id : 640653566375 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Which of the following statements is false regarding webhooks?

**Options :**

6406531892863. ✖ A webhook should deliver data to other apps, as it happens.

6406531892864. ✖ A webhook is primarily meant for a server to server communication.

6406531892865. ✖ A webhook typically uses an HTTP POST request to deliver the response.

6406531892866. ✔ A webhook and web socket is essentially the same.

**Question Number : 403 Question Id : 640653566382 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Suppose an endpoint in your flask application triggers a celery task, which generates a CSV file named "data.csv" and saves it in the static folder of a flask application. The file generator takes anywhere between 20 and 40 seconds to generate and save the CSV file.

The below fetch call is used to get the file generated by the above explained celery task.

```
fetch("/static/data.csv").then(response => response.blob())
).then(data => {
  // does something with the response data
})
```

Which of the following is the most efficient way to make the above fetch call so that it doesn't fail and return the desired response?



**Options :**

6406531892891. ✔ Use short polling to check the state of the celery task after every 5 seconds, and make the fetch call, when the task succeeds.
6406531892892. ✖ Use javascript function “setTimeout” to make the fetch after 41 seconds, which makes sure that the file is generated and saved to the desired location.

Sub-Section Number :	6
Sub-Section Id :	64065381009
Question Shuffling Allowed :	Yes
Is Section Default? :	null

**Question Number : 404 Question Id : 640653566370 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the following Vue application with markup “index.html” and javascript file “app.js”.

index.html:

```
<div id = "app">
  <my-comp>
    <template #header>
      This is header content
    </template>

    <template #footer>
      This is footer content
    </template>

    <p> This is some content </p>
  </my-comp>
</div>
<script src = "app.js"> </script>
```

app.js:

```
Vue.component("myComp", {
  template : `<div>
    <p>
      <slot name="header"></slot>
    </p>
    <p>
      <slot></slot>
    </p>
  </div>
`,
})

const app = new Vue({
  el : "#app",
})
```

Suppose you open the "index.html" file in a browser. What will be rendered by the browser?

**Options :**

This is header content

This is footer content

6406531892843. ✖ This is some content

6406531892844. ✓ This is header content  
This is some content

6406531892845. ✖ This is header content

6406531892846. ✖ This is some content  
This is header content

**Question Number : 405 Question Id : 640653566371 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the following javascript program, and predict the output if executed.

```

new Promise((str1, str2) => {
  if (5 == "5") str1(5)
  else str2(8)
}).
then(d => {
  console.log("Checkpoint 4", d);
  throw new Error(20);
  return d * 5;
})
.then(d => {
  console.log("Checkpoint 2", d);
  return d;
}, d => {
  console.log("Checkpoint 1", d.message);
  return d.message * 2;
}).catch(e => {
  console.log("Checkpoint 3", e.message);
  return e.message * 2;
}).finally(d => {
  console.log("Checkpoint 6", d);
  return d * 5;
}).then(d => {
  console.log("Checkpoint 5", d);
  return d * 5;
})

```

Options :

Checkpoint 4 5  
 Checkpoint 3 20  
 Checkpoint 1 undefined  
 Checkpoint 6 NaN

6406531892847. ✖

Checkpoint 4 8  
 Checkpoint 1 20  
 Checkpoint 6 NaN  
 Checkpoint 5 40

6406531892848. ✖

Checkpoint 4 5  
 Checkpoint 1 20  
 Checkpoint 6 undefined  
 Checkpoint 5 40

6406531892849. ✔

Checkpoint 4 5

Checkpoint 5 20

Checkpoint 1 40

6406531892850. ✖ Checkpoint 6 200

**Question Number : 406 Question Id : 640653566372 Question Type : MCQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the following Vue application with markup "index.html" and javascript file "app.js".

index.html:

```
<div id = "app">
  <input v-model = "text" @input = "save_data">
  <p> {{content}} </p>
</div>
<script scr = "app.js"></script>
```

app.js:

```
new Vue({
  el : "#app",
  data : {
    text : "",
    content : "",
  },

  mounted () {
    try {
      this.text = localStorage.getItem("value1").split("abhi").slice(1,
-1),join("");
      this.content =
localStorage.getItem("value1").split("abhi").slice(1),join("");
    }
    catch {
      this.text = "";
      this.content = "";
    }
  },

  methods : {
    save_data() {
      localStorage.setItem("value1", this.text);
      localStorage.setItem("value2", this.content);
    }
  }
})
```

Suppose you open "index.html" file in a browser, and type the text "abhishekabhinavabhi" in the text box shown (after removing the previous text, if any), and hard refresh the page twice, without clicking anywhere. What will be the value shown in the text box, and the "age" placeholder, respectively?

**Options :**

6406531892851.

✓ sheknav, sheknav

6406531892852. ✖ nav, nav

6406531892853. ✖ shek, sheknav

6406531892854. ✖ <sup>let</sup> , nav

**Question Number : 407 Question Id : 640653566378 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Which of the following shows the correct output if the javascript program written below is executed?

```
let num = 50

const a = {
  num : 10,
  func : function (num) {
    console.log("Function A:", num)
  }
}

const b = {
  num : 20,
  func : function () {
    console.log("Function B:", this.num)
    a.func.bind(this)(num = 40)
  }
}

b.func.apply(a, [40])
```

**Options :**

6406531892875. ✖ Function B: 20  
Function A: 10

6406531892876. ✖ Function B: 40  
Function A: 40

6406531892877. ✖ Function B: 10  
Function A: 10

6406531892878. ✔ Function B: 10  
Function A: 40

**Question Number : 408 Question Id : 640653566388 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the below application with markup "index.html" and javascript file "app.js".



index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
new Vue({
  el: '#app',
  template: `<div>
<ol type='1'>
  <li v-for='fruit in fruits'>{{fruit}}</li>
</ol>
</div>`,
  data: {
    fruits: ['Banana', 'Mango'],
  },
  created() {
    this.fruits.push('Orange')
  },
  mounted() {
    this.fruits.push('Apple')
  },
})
```

Suppose you open the file "index.html" in a browser. What will be rendered by the browser?

**Options :**

1) Banana

6406531892913. ✖ 2) Mango

1) Banana

2) Mango

6406531892914. ✖ 3) Orange

6406531892915. ✖

- 1) Banana
- 2) Mango
- 3) Apple

6406531892916. ✔

- 1) Banana
- 2) Mango
- 3) Orange
- 4) Apple

**Question Number : 409 Question Id : 640653566389 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the below application with markup "index.html" and javascript file "app.js".

index.html:

```
<body>
  <div id="app"></div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script src="app.js" type="module"></script>
</body>
```

app.js:

```
const Player = {
  template: `<div style='background-color:red'><slot></slot></div>`,
  data() {
    return {
      scores: [
        { name: 'Rohit', run: 20 },
        { name: 'Virat', run: 50 },
      ],
    }
  },
}

new Vue({
  el: '#app',
  template: `<div>
    <h1> Score Board</h1>
    <Player>
      <ol type='1'>
        <li v-for='score in scores'>{{score.name}}:
{{score.run}}</li>
      </ol>
    </Player>
  </div>`,
  data: {
    scores: [
      { name: 'Rohit', run: 50 },
      { name: 'Virat', run: 20 },
    ],
  },
  components: {
    Player,
  },
})
```

Suppose you open the file “index.html” in a browser. What will be rendered inside the slot, by the browser?

**Options :**

1) Rohit: 20

6406531892917. ✖ 2) Virat: 50

6406531892918. ✓  
1) Rohit: 50  
2) Virat: 20

6406531892919. ✖  
1) Rohit: 50  
2) Virat: 50

6406531892920. ✖  
1) Rohit: 20  
2) Virat: 20

**Question Number : 410 Question Id : 640653566396 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the below flask application.

app.py:

```
from flask import Flask
from flask_caching import Cache
from time import sleep

config = {
    "CACHE_TYPE": "RedisCache",
    "CACHE_REDIS_URL": 'redis://localhost:6379/1'
}

app = Flask(__name__)
app.config.from_mapping(config)
cache = Cache(app)

@cache.memoize(timeout=120)
def get_score(name):
    sleep(10)
    return 0

@app.route('/match/<int:match_id>/player/<name>/score')
def score(match_id, name):
    score = get_score(name)
    return f"IND Vs AUS- SKY: {score}"

if __name__ == '__main__':
    app.run(debug=True)
```

If the Redis server is running on "localhost:6379", and application is running on "<http://127.0.0.1:5000>". If a user visits the URL "<http://127.0.0.1:5000/match/1/player/sky/score>" at 7AM in the morning for the first time and then at 9AM for the second time. What will be the approximate absolute difference between the response time of the requests?

**Options :**

6406531892945. ✖ 10 seconds

6406531892946. ✔ 0 seconds

6406531892947. ✖ 20 seconds

6406531892948. ✖ 120 Seconds

**Question Number : 411 Question Id : 640653566397 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the below flask application.

app.py:

```
from flask import Flask
from flask_caching import Cache
from time import sleep

config = {
    "CACHE_TYPE": "RedisCache",
    "CACHE_REDIS_URL": 'redis://localhost:6379/1'
}

app = Flask(__name__)
app.config.from_mapping(config)
cache = Cache(app)

@cache.memoize(timeout=120)
def get_score(name):
    sleep(10)
    return 0

@app.route('/match/<int:match_id>/player/<name>/score')
def score(match_id, name):
    score = get_score(name)
    return f"IND Vs AUS- SKY: {score}"

if __name__ == '__main__':
    app.run(debug=True)
```

If the Redis server is running on "localhost:6379", and application is running on "<http://127.0.0.1:5000>". If a user visits the URL "<http://127.0.0.1:5000/match/1/player/sky/score>" first and then "<http://127.0.0.1:5000/match/2/player/sky/score>" within 2 minutes. What will be the approximate absolute difference between the response time of the requests?

**Options :**

6406531892949. ✓ 10 seconds

6406531892950. ✖ 0 seconds

6406531892951. ✖ 20 seconds

6406531892952. ✖ 120 Seconds

## MLF

Section Id :	64065338412
Section Number :	15
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065381010
Question Shuffling Allowed :	No
Is Section Default? :	null

**Question Number : 412 Question Id : 640653566398 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 0**

Question Label : Multiple Choice Question

**THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING FOUNDATIONS (COMPUTER BASED EXAM)"**

**ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?**

**CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.**