Week 7 Graded Questions

Q1: Consider the following markup index.html and JavaScript app.js for a Vue application.

index.html:

```
<div id="app">
      <div>
        <router-link :to="{name:'home'}"> Home </router-link>
        <router-link :to="{name:'profile', params:{name:'Abhisek'}}">
          Profile</router-link
        >
      </div>
      <router-view />
    </div>
<script src="app.js"></script>
```

app.js:

```
const home = {
  template: `<h1> Home </h1>`,
}

const profile = {
  template: `<h1> Hello {{ name }}</h1>`,
  props: ['name'],
}

const routes = [
  { path: '/', name: 'home', component: home },
  { path: '/profile/:name', name: 'profile', props: true, component:
profile },
]

const router = new VueRouter({
  routes,
  base: '/',
})

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

What will be rendered inside the router-view when user loads the application for the first time and when user clicks on the link “Profile", respectively?

A. Blank Page, Hello Abhisek
B. Hello Abhisek, Home
C. Home, Hello Abhisek
D. None of the above

Answer: C

Solution: Because base property of router instance is ‘/’. A component associated with ‘/’ which is home will be rendered inside the router-view component. Once the user clicks on the Profile, the user will be redirected to the route with the name ‘profile’ along with the parameter name = Abhisek. props = true so, parameters will be passed as props. So, answer C is correct.

Q2: Consider the following markup index.html and JavaScript app.js for a Vue application.

app.js:

```
const player = {
  template: `<div>
    <h1> Name: {{ name }}</h1>
    <router-view />
  </div>`,
  props: ['name'],
}

const test = {
  template: '<div><h1> Test Runs: {{ runs }} </h1></div>',
  data() {
    return { runs: 5000 }
  },
}

const oneDay = {
  template: '<div><h1> Oneday Runs: {{ runs }} </h1></div>',
  data() {
    return { runs: 10000 }
  },
}

const routes = [
  {
    path: '/player/:name',
    component: player,
    children: [
      { path: '', component: oneDay },
      { path: 'test', component: test },
      { path: 'oneday', component: oneDay },
    ],
    props: true,
  },
  { path: '*', component: test },
]
const router = new VueRouter({
  routes,
  base: '/',
})

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

index.html:

```html
<div id="app">
   <router-view></router-view>
</div>
<script src="app.js"></script>
```

Suppose the application is running on http://localhost:8080, and user visits the URL "http://localhost:8080/#/player/rohit". What will be rendered inside the router-view?

A. Name: Rohit
   Oneday Runs: 10000

B. Test Runs: 5000

C. Name: Rohit
   Test Runs: 5000

D. Oneday Runs: 10000

Answer: A

Solution: Because nothing has been mentioned after the "http://localhost:8080/#/player/rohit". The component associated with path ' ' inside the children list will be rendered inside the router-view of player component template. Which is oneDay so, Option A.

Q3: Consider the app.js and index.html given in the Question No.2.

Suppose the application is running on http://localhost:8080, and the user visits the URL "http://localhost:8080/#/player/rohit/test". What will be rendered by the browser?

A. Name: Rohit
   Oneday Runs: 10000

B. Test Runs: 5000

C. Name: Rohit
   Test Runs: 5000

D. Oneday Runs: 10000

Answer: C

Solution: Because '/test' is after the "http://localhost:8080/#/player/rohit". The component associated with path 'test' inside the children list will be rendered inside the router-view of player component template, which is "test" so, Option C.

Q4: Consider the app.js and index.html given in the Question No.2.

Suppose the application is running on http://localhost:8080, and the user visit the URL "http://localhost:8080/#/player/rohit/t20". What will be rendered by the browser?

A. Name: Rohit
   Oneday Runs: 10000

B. Test Runs: 5000

C. Name: Rohit
   Test Runs: 5000

D. Oneday Runs: 10000

Answer: B

Solution: Because '/t20' is after the "http://localhost:8080/#/player/rohit". So, router will not be able to find any match. So, the component associated with '*' will be rendered inside the router-view of div with ID "app".

Q5: Consider the following markup index.html and JavaScript app.js for a Vue application.

index.html:

```
<div id="app">
    <div>
        <router-link to="/profile" replace> Profile </router-link>
        <router-link to="/post"> Post </router-link>
    </div>
    <div>
```

```
        <router-view />
    </div>
</div>
```

app.js:

```
const profile = {
  template: '<div> <p> Profile Page </p> </div>',
}

const post = {
  template: '<div><p> Users Post </p></div>',
}

const routes = [
  { path: '/profile', component: profile },
  { path: '/post', component: post },
  { path: '*', component: post },
]
const router = new VueRouter({
  routes,
  base: '/',
})

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

Suppose the application is running on http://localhost:8080. If after running the application in new tab for the first time (browser's history stack is empty), and the user clicks on the link 'Profile' then which of the following statement is true?

A. 'Profile Page' will be rendered inside router-view, the user will not be able to navigate back to the default post page using browser's back button.
B. 'Profile Page' will be rendered inside router-view, the user will be able to navigate back to the default post page using browser's back button.

C. ‘Users Post’ will be rendered inside router-view, the user will not be able to navigate back to the default post page using browser’s back button.
D. ‘Users Post’ will be rendered inside router-view, the user will be able to navigate back to the default post page using browser’s back button.

Answer: A

Solution: Once the user visit the app for first time, post component will be rendered inside the router view and ‘/post’ will be pushed inside the browser’s history stack. When the user click on ‘Profile’, profile component will be rendered inside the router-view and ‘/post’ will be replaced with the ‘/profile’ because of replace props so, the browser’s history stack will look something like [‘/profile’] so, user will not be able to go back to default post page using browsers back button. So, option A.

Q6: Consider app.js and index.html from the Question No.5.

Suppose the application is running on http://localhost:8080. If after running the application in new tab for the first time (browser’s history stack is empty), and the user clicks on the link “Profile” and then ‘Post’, then which of the following statement is true?

A. ‘Profile Page’ will be rendered inside router-view, the user will not be able to navigate back to the default post page using browser’s back button.
B. ‘Profile Page’ will be rendered inside router-view, the user will be able to navigate back to the default post page using browser’s back button.
C. ‘Users Post’ will be rendered inside router-view, the user will not be able to navigate back to the default post page using browser’s back button.
D. ‘Users Post’ will be rendered inside router-view, the user will be able to navigate back to the default post page using browser’s back button.

Answer: C

Solution: When the user visits the app for the first time, the default ‘/post’ will be pushed to the browser’s history stack. And then when the user clicks on the ‘profile’ link, ‘/post’ will be replaced with the ‘/profile’ because of the ‘replace’ prop. Again, when the user visits the post page, ‘/post’ will be pushed to the history stack. So at the end stack will look something like [‘/profile’, ‘/post’] so, user will not be able to navigate back to the default ‘post’ page using browsers back button.

Q7: In the code snippet below, the Vuex store has a state called product.

Fill in the code in ***code1*** inside the “products” computed property in order to render the name and prices of the different products.

index.html:

```
<div id = "app">
         <table border="1px">
           <tr>
                <td>Product</td>
                <td>Price</td>
           </tr>
           <tr v-for="product in products">
               <td>{{product.name}}</td>
               <td>{{product.price}}</td>
           </tr>
       </table>

    </div>
```

app.js:

```

const store = new Vuex.Store({
    state: {
      count:0,
      products:[
        {name:'apple',
        price:120
        },
        {
          name:'banana',
          price:60
        }
      ]
    },


  })
   new Vue({
    el:"#app",
    store:store,

    computed:{
        products(){
            return code1
          }
    }
})
```

A. store.commit.products
B. store.state.products
C. store.dispatch.products
D. None of the above

Answer: B

Solution: One can retrieve the values in the central store by returning the store state within a computed property. In the above example, it should be store.state.products

Q8: Fill in ***code1***, inside a mutation handler, which can be used to reduce the price of each of the products in the state by a certain discount.

app.js:

```
const store = new Vuex.Store({
    state: {
      discount:10,
      products:[
        {name:'apple',
        price:120
        },
        {
          name:'banana',
          price:60
        }
      ]
    },
  mutations: {
      reducePrice:state=>{
          Code 1

      }
    }
})
```

A. this.$store.state.products.forEach(product => {
   product.price -= (state.discount)/100*product.price;})

B. state.products.forEach(product => {
   product.price -= (state.discount)/100*product.price;})

C. commit.products.forEach(product => {
   product.price -= (state.discount)/100*product.price;})

D. dispatch.products.forEach(product => {
   product.price -= (state.discount)/100*product.price;})

Answer: B

Solution: Correct option is B. The mutation handler receives state as its first argument. The discount needs to be applied on each of the products.

Q9: Which of the following statement(s) is are correct?

A. Web Manifests, service workers are some characteristics of a PWA.
B. All SPAs are PWAs.
C. A Web worker is a script started by a web content that runs in the background, which can perform computations and fetch requests and communicate back messages to the origin web content.
D. Search engine optimization, managing browser history are some challenges of SPAs.

Answer A, C and D

Solution: Reference :
https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction

Q10: Which of the following statements is false regarding Vuex?

A. The mutations must be synchronous.
B. The state provided by Vuex store is shared among all the components of the application.
C. An action should not directly change the state of a Vuex store, and must call a mutation to change the state.
D. Only the direct child components (not descendants) of the app have the access to the “this.$store” variable.

Answer: D

Solution: All components of an application have access to the centralized store.