

# COS30043 – Video Coding Tutorial

Task 10.2HD High Distinction Project



Alliance with  Education

Name: Le Gia Hoang An

Student ID: 104789808

Unit: COS30043

This report is for documenting evidence of the video tutorial for Task 10.2HD High Distinct Project. The tutorial will be covering topics from the lecture slides in week 4, week 6, and week 7. All these topics are implemented in the custom web application project, Task 6.2D-HD. For this project, this will be video coding of the login page that was built in the customer web application project, Task6.2D-HD. This project uses VueJS and Bootstrap frameworks, PHP fetch API, and MySQL database. In addition to being able to perform insert data into the MySQL database, a method of form validation was also implemented to ensure that only data under the correct format or filled with data input will be accepted, before it was being set to the Fetch API.

All code will that is written in the video tutorial will be included in the compressed ZIP file.

VueJS file and Bootstrap CDN include:

The screenshot shows the Vue.js documentation page. The main content area is titled 'Using Vue from CDN'. It explains that you can use Vue directly from a CDN via a script tag. A code block shows the script tag: `<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>`. It then explains that here they are using unpkg, but you can also use any CDN that serves npm packages, for example jsdelivr or cdnjs. It also mentions that when using Vue from a CDN, there is no "build step" involved. Below this, it shows 'Using the Global Build' with a code block that demonstrates how to use the global Vue object to create a Vue application. The code block includes the script tag and the Vue application setup. The right sidebar has a section 'ON THIS PAGE' with links to 'Try Vue Online', 'Creating a Vue Application', 'Using Vue from CDN', and 'Next Steps'. There is also a 'Become a Sponsor' link at the bottom of the sidebar.

3. **Hello, world!** Open the page in your browser of choice to see your Bootstrapped page. Now you can start building with Bootstrap by creating your own [layout](#), adding dozens of [components](#), and utilizing [our official examples](#).

## CDN links

As reference, here are our primary CDN links.

Description	URL
CSS	<a href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css</a>
JS	<a href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js</a>

You can also use the CDN to fetch any of our [additional builds listed in the Contents page](#).

## VueJS Component Code

```
js > components > login.js > Login
1  const Login = {
2    template: `
3      <form v-on:submit = "checkForm">
4        <label for="username" class="form-label">Username</label>
5        <input v-model="input.username" type="text" id="username" class="form-control">
6        <p class="text text-danger" v-for="error in usernameErrors">{{error}}</p>
7
8        <label for="password" class="form-label">Password</label>
9        <input v-model="input.password" type="password" id="password" class="form-control">
10       <p class="text text-danger" v-for="error in passwordErrors">{{error}}</p>
11
12       <button type="submit" class="btn btn-primary">Login</button>
13
14       <p class="text text-danger">{{message}}</p>
15     </form>
16   `,
17   data() {
18     return {
19       input: {
20         username: "",
21         password: "",
22       },
23       message: "",
24       usernameErrors: [],
25       passwordErrors: [],
26     };
27   },
28   methods: {
29     login() {
30       const formData = new FormData();
31       formData.append("username", this.input.username);
32       formData.append("password", this.input.password);
33
34       const requestOptions = {
35         method: "POST",
36         body: formData
37       }
```

```
loginjs  X  index.html  script.js  apis.php  style.css
js > components > loginjs > Login
1  const Login = {
29  methods: {
30    login() {
35      const requestOptions = {
37        body: formData,
38      };
39
40      fetch("resources/apis.php?action=login", requestOptions)
41        .then((response) => {
42          if (response.status === 202) {
43            this.$root.authenticatedUser = this.input.username;
44            this.$root.setAuthenticated(true);
45            this.$router.replace({ name: "dashboard" });
46          } else if (response.status === 501) {
47            this.message = "Invalid login credentials";
48          }
49        })
50        .catch((error) => {
51          console.log("Error: " + error);
52        });
53    },
54    checkForm(event) {
55      this.usernameErrors = [];
56      this.passwordErrors = [];
57      let result = true;
58
59      if (!this.input.username) {
60        this.usernameErrors.push("Username is required");
61        result = false;
62      }
63
64      if (!this.input.password) {
65        this.passwordErrors.push("Password is required");
66        result = false;
67      }
68
69      if (result === true) {
```

```
loginjs  X  index.html  script.js  apis.php  style.css
js > components > loginjs > Login
1  const Login = {
29  methods: {
54    checkForm(event) {
64      if (!this.input.password) {
65        this.passwordErrors.push("Password is required");
66        result = false;
67      }
68
69      if (result === true) {
70        this.login();
71      } else {
72        event.preventDefault();
73      }
74    },
75  },
76 };
77
```

## PHP Backend Code

```
login.js index.html script.js apis.php style.css
resources > apis.php
6
7 // connect to the mysql database, provide the appropriate credentials
8 $conn = mysqli_connect('localhost', 'root', '', 'realestate');
9 mysqli_set_charset($conn, 'utf8');
10 $response = array();
11 $i = 0;
12
13 if ($conn) {
14     switch ($_GET['action']) {
15         //////////////////////////////////////////////////Login function////////////////////////////////////
16         case 'login':
17             if ($_SERVER['REQUEST_METHOD'] === "POST") {
18                 $username = $_POST['username'];
19                 $password = $_POST['password'];
20
21                 $sql = "SELECT * FROM users WHERE UserName = ? AND Password = ?";
22                 $stmt = $conn->prepare($sql);
23                 $stmt->bind_param("ss", $username, $password);
24                 $stmt->execute();
25
26                 $result = $stmt->get_result();
27                 $login = $result->fetch_assoc();
28
29                 if (isset($login['UserName']) && isset($login['Password'])) {
30                     http_response_code(202);
31                 } else {
32                     http_response_code(501);
33                 }
34
35                 $stmt->close();
36
37             } else {
38                 http_response_code(405);
39             }
40             break;
41     }
```

Video Tutorial Link:

<https://youtube.com/watch?v=waMOxwXqxjE>