



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ Ura.
Nom d'usage ▶ Ura.
Prénom ▶ Mersad.
Adresse ▶ 52 Avenue Raymond Bergougnan.

Titre professionnel visé

CONCEPTEUR DÉVELOPPEUR D'APPLICATIONS

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

p.
.

► Le Juste Prix

p. p. 7

► RESTons connecté

p. p. 20

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

p.
.

► Cool Raoul

p. p. 12

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

p.
.

► Ca gratte

p. p. 28

► Flask test

p. p. 36

Intitulé de l'activité-type n° 4

p.
.

Titres, diplômes, CQP, attestations de formation *(facultatif)*

p. 42

Déclaration sur l'honneur

p. 43

Documents illustrant la pratique professionnelle *(facultatif)*

p. 44

Annexes *(Si le RC le prévoit)*

p. 45



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

DOSSIER PROFESSIONNEL^(DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Exemple n°1 ▶

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

LE JUSTE PRIX

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

LE JUSTE PRIX

Le projet était fait dans le cas de brief avec simplon.co, pour valider des compétences comme:

- Développer une interface utilisateur de type desktop
- Développer la partie front-end d'une interface utilisateur web.

Le projet est une application où l'utilisateur doit deviner le chiffre. L'application choisit aléatoirement un chiffre entre 0 et 50 et l'utilisateur doit deviner en 5 coups max le chiffre. L'application a des features dark/light mode.

On a mis en place le fichier pour configurer électron qui va permettre d'afficher l'application sur l'app desktop. La fonction create window va afficher le jeu en mode desktop. Ensuite on a créé la logique du jeu. La fonction checkGuess va comparer le numéro entré avec le numéro caché, si c'est le même on affiche un message de quoi l'utilisateur a gagné.

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Technologie utilisée: Électron pour développer l'application sur desktop. Javascript pour le mécanisme du jeu. Css pour le style.

3. Avec qui avez-vous travaillé ?

J'ai travailler avec Jordy Bastide, Marvin Silva

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service	▶	Le juste prix
Période d'exercice	▶	Du 09.05.2022 au 13.05.2022

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

```
1 const { app, BrowserWindow, ipcMain, nativeTheme } = require('electron');
2 const path = require('path');
3 require("electron-reload")(__dirname);
4
5 const createWindow = () => {
6   const win = new BrowserWindow({
7     width: 800,
8     height: 600,
9     webPreferences: {
10       preload: path.join(__dirname, 'preload.js')
11     }
12   })
13
14   win.loadFile('index.html')
15
16   ipcMain.handle('dark-mode:toggle', () => {
17     if (nativeTheme.shouldUseDarkColors) {
18       nativeTheme.themeSource = 'light'
19     } else {
20       nativeTheme.themeSource = 'dark'
21     }
22     return nativeTheme.shouldUseDarkColors
23   })
24
25   ipcMain.handle('dark-mode:system', () => {
26     nativeTheme.themeSource = 'system'
27   })
28 }
29
30 app.whenReady().then(() => {
31   createWindow()
32
33   //Open a window if none are open on macos
34   app.on('activate', () => {
35     if (BrowserWindow.getAllWindows().length === 0) createWindow()
36   })
37 })
38
39 app.on('window-all-closed', () => {
40   if (process.platform !== 'darwin') app.quit()
41 })
```

La logique du jeu.

DOSSIER PROFESSIONNEL (DP)

```
function checkGuess() {
  const userGuess = Number(guessField.value);
  if (guessCount === 1) {
    guesses.textContent = "Choix précédents: ";
  }

  if ((userGuess <= 50) & (userGuess >= 0)) {
    guesses.textContent += userGuess + " ";

    if (userGuess === randomNumber) {
      lastResult.textContent = "Bravo! Vous avez trouvé!";
      lastResult.style.backgroundColor = "green";
      lowOrHi.textContent = "";
      setGameOver();
    } else if (guessCount === 5) {
      lastResult.textContent = "! GAME OVER !";
      lowOrHi.textContent = "";
      setGameOver();
    } else {
      lastResult.textContent = "Perdu!";
      lastResult.style.backgroundColor = "red";
      if (userGuess < randomNumber) {
        lowOrHi.textContent = "Trop bas!";
      } else if (userGuess > randomNumber) {
        lowOrHi.textContent = "Trop haut!";
      }
    }
  }

  guessCount++;
  guessField.value = "";
  guessField.focus();
} else {
  alert("Vous devez choisir un chiffre entre 0 et 50");
}
}

document.addEventListener("DOMContentLoaded", function () {
  guessSubmit.addEventListener("click", checkGuess);
});
```

DOSSIER PROFESSIONNEL (DP)

```
document.addEventListener("DOMContentLoaded", function () {
  guessSubmit.addEventListener("click", checkGuess);
});

function setGameOver() {
  guessField.disabled = true;
  guessSubmit.disabled = true;
  resetButton = document.createElement("button");
  resetButton.textContent = "Rejouer";
  resetButton.classList.add("btn", "btn-primary");
  appendReset.appendChild(resetButton);
  resetButton.addEventListener("click", resetGame);
}

function resetGame() {
  guessCount = 1;
  const resetParas = document.querySelectorAll(".resultParas p");
  for (const resetPara of resetParas) {
    resetPara.textContent = "";
  }

  resetButton.parentNode.removeChild(resetButton);
  guessField.disabled = false;
  guessSubmit.disabled = false;
  guessField.value = "";
  guessField.focus();
  lastResult.style.backgroundColor = "white";
  randomNumber = Math.floor(Math.random() * 50) + 1;
}

module.exports = randomNumber;
```

Activité-type 2

Exemple n°1 ▶

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Cool Raoul

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Cool Raoul

Le projet était fait dans le cas de brief avec simplon.co, pour valider des compétences comme:

- Maquette d' une application.
- Développer une application mobile.
- Concevoir une application.

Le projet est une application de planification des afterworks . L'application permettant aux entreprises de mieux organiser leurs afterworks.

Pour cela, les entreprises peuvent créer un groupe au sein duquel elles pourront inviter l'ensemble de leurs collaborateurs.

Les collaborateurs ont ainsi accès à un "calendrier" commun où chacun pourra créer un ou plusieurs afterwork(s) quand il ou elle le souhaitera. Pour chaque évènement, un chat intégrant tous les participants est généré.

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Technologie utilisée: React pour le front. Ts. Css pour le style.

3. Avec qui avez-vous travaillé ?

Je travaille avec l'ensemble de la promo.

4. Contexte

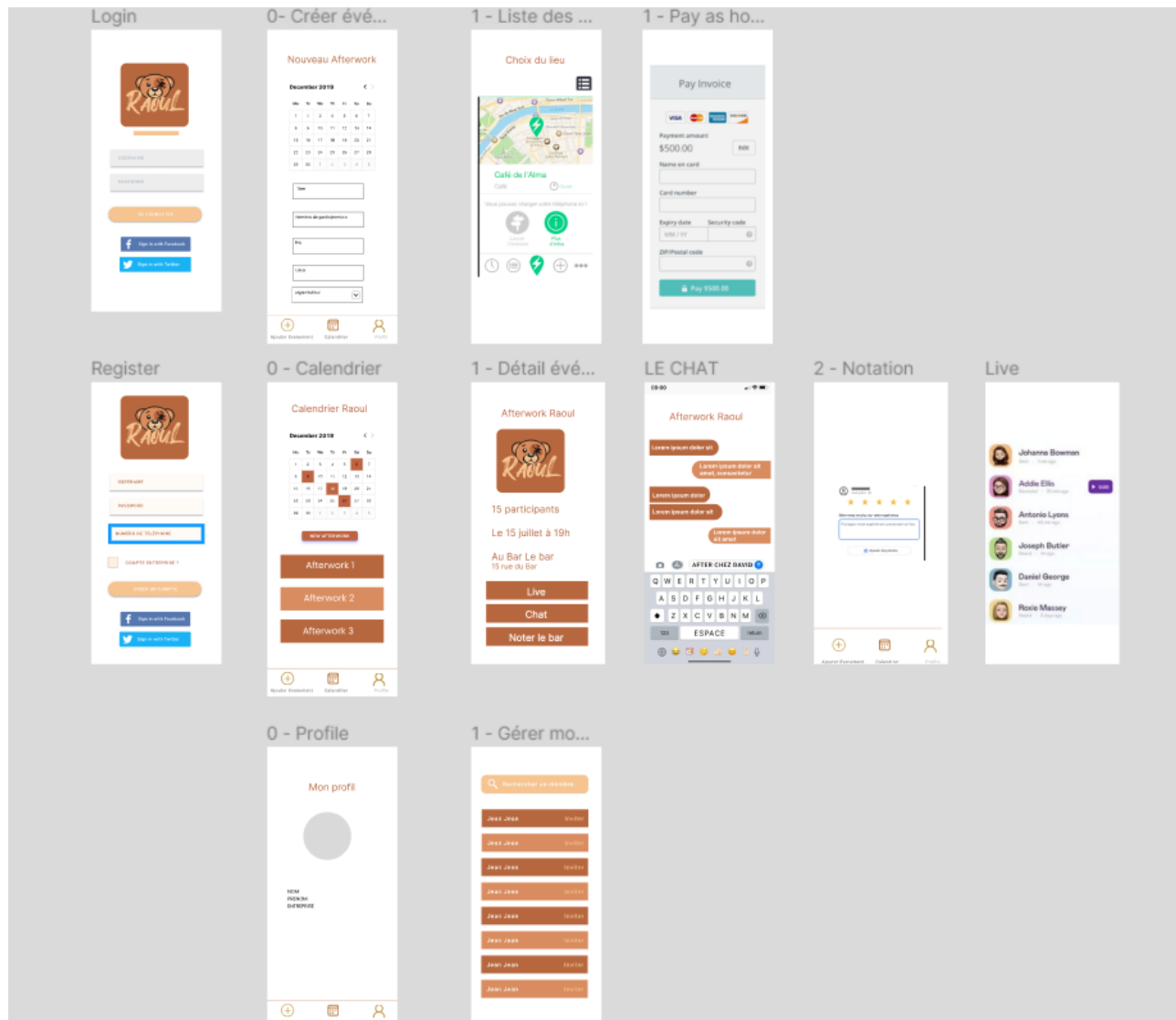
Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service	▶	Cool Raoul
Période d'exercice	▶	Du 07.06.2022 au 09.06.2022

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Maquette



User entity

DOSSIER PROFESSIONNEL (DP)

```
import { DataTypes, Model } from "sequelize";
import { sequelize } from "../database";

export class User extends Model {
  declare id?: number;
  declare firstName: string;
  declare lastName: string;
  declare phoneNumber: string;
  declare password: string;
}

User.init(
  {
    id: {
      type: DataTypes.INTEGER,
      autoIncrement: true,
      primaryKey: true,
    },
    firstName: {
      type: DataTypes.STRING,
      allowNull: true,
    },
    lastName: {
      type: DataTypes.STRING,
      allowNull: true,
    },
    phoneNumber: {
      type: DataTypes.STRING,
      allowNull: true,
      unique: true,
    },
    password: {
      type: DataTypes.STRING,
      allowNull: true,
    },
  },
  {
    sequelize,
    modelName: "users",
  }
);
```

Event Entity

```
import { DataTypes } from "sequelize";
import { sequelize } from "../database";

export const Event = sequelize.define("events", {
  id: {
    type: DataTypes.INTEGER,
    autoIncrement: true,
    primaryKey: true,
  },
  title: {
    type: DataTypes.STRING,
  },
  capacity: {
    type: DataTypes.INTEGER,
  },
  price: {
    type: DataTypes.INTEGER,
  },
  date: {
    type: DataTypes.DATEONLY,
    allowNull: false,
  },
  startTime: {
    type: DataTypes.TIME,
  },
  endTime: {
    type: DataTypes.TIME,
  },
});
```




MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Front

DOSSIER PROFESSIONNEL (DP)

```
export default function Calendar({ navigation }) {  
  let [events, setEvents] = useState([]);  
  [events, setEvents] = GetAllEvents([events, setEvents]);  
  
  const eventBtns = events.map((item, index) => {  
    return (  
      <ButtonApp  
        key={index}  
        eventTitle={item.title}  
        eventDate={item.date}  
        eventStart={item.startTime}  
        eventEnd={item.endTime}  
        color={COLORS.primary}  
        width="100%"  
        typo={FONTS.h2_semiBold}  
        padding={20}  
        onPress={() =>  
          clickBtn(item.title, item.date, item.startTime, item.endTime)  
        }  
      />  
    );  
  });  
  
  return (  
    <ScrollView contentContainerStyle={styles.container}>  
      <Text style={styles.title}>Calendrier Raoul</Text>  
      <CalendarApp events={events} />  
      <ButtonApp  
        eventTitle="new after work"  
        color={COLORS.primary}  
        width="60%"  
        typo={FONTS.h3_semiBold}  
        padding={10}  
        marginBottom={10}  
        onPress={() => navigation.navigate("NewEvent", { barId: {} })}  
      />  
      {eventBtns}  
    </ScrollView>  
  );  
}
```

APP.JSX

DOSSIER PROFESSIONNEL (DP)

```
const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen
          name="SignIn"
          component={SignInScreen}
          options={{ title: "Sign in" }}
        />
        <Stack.Screen
          name="Home"
          component={HomeScreen}
          options={{ title: "Welcome" }}
        />
        <Stack.Screen
          name="NewEvent"
          component={NewEvent}
          options={{ title: "New Event" }}
        />
        <Stack.Screen
          name="Chat"
          component={ChatScreen}
          options={{ title: "Afterwork Raoul" }}
        />
        <Stack.Screen
          name="Profil"
          component={ProfileScreen}
          options={{ title: "Profil" }}
        />
        <Stack.Screen
          name="Group"
          component={GroupScreen}
          options={{ title: "Gérer mon gang" }}
        />
        <Stack.Screen
          name="Event"
          component={Event}
          options={{ title: "Détail de l'événement" }}
        />
        <Stack.Screen
          name="Calendar"
          component={Calendar}
          options={{ title: "Calendar" }}
        />
        <Stack.Screen
          name="Map"
          component={MapScreen}
          options={{ title: "Map" }}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

Activité-type 1

Exemple n°2 ▶

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

RESTons connecté

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

RESTons connecté:

Le projet était fait dans le cas de brief avec simplon.co, pour valider des compétences comme:

- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web.
- Développer la partie back-end d'une interface utilisateur web.
- Concevoir une application.

Le projet est une application où on met en avant les données disponibles de tous les pays du monde pour tout le monde. On a aussi mis en place à la fois une API Rest mais également une page HTML/CSS/JS afin d'effectuer des recherches via un formulaire qui retournera les résultats. Le modèle "country" contiendra au moins un code qui doit être sur 2 caractères en majuscule et un nom en chaîne de caractères. En place une validation des champs.

J'ai commencé par installer Java dans mon pc, ensuite j'ai initialisé spring boot, dans <https://start.spring.io/>, avec les dépendances, comme mysql, validator, data jpa. Ensuite j'ai connecté avec mysql dans application.properties.

J'ai créé la table Country dans entity/Country.java pour les propriétés de la table.

Le CountryRepository dans repository/CountryRepository.java pour le CRUD.

Le MainController dans controller/MainController.java.

Dans / on trouve index avec le formulaire pour ajouter, modifier une voiture et regarder tout le pays.

J'ai run springboot avec la commande `./mvnw spring-boot:run`

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Technologie utilisée: Mysql pour la base de donner. Javascript pour le mécanisme du jeu. Css pour le style. Springboot.

3. Avec qui avez-vous travaillé ?

J'ai travailler avec Allan Gagne.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service ▶ RESTons connecté

Période d'exercice ▶ Du 07.12.2021 au 24.12.2021

5. Informations complémentaires (facultatif)

```
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private Integer id;

@NotNull(message = "Ne doit pas être vide ")
@Pattern(regexp = "[A-Za-z]*", message = "Doit avoir que des caractères")
private String name;
@NotNull
@Size(min = 2, max = 2, message = "Doit contenir que 2 caractères")
@Pattern(regexp = "[A-Za-z]*", message = "Doit avoir que des caractères")
private String code;

public Country(String name, String code) {
    this.name = name;
    this.code = code;
}

public Country() {
}

public void setId(Integer id) {
    this.id = id;
}

public Integer getId() {
    return id;
}

public void setName(String name) {
    this.name = name;
}

public String getName() {
    return name;
}

public void setCode(String code) {
    this.code = code.toUpperCase();
}

public String getCode() {
    return code;
}
}
```

DOSSIER PROFESSIONNEL (DP)

```
public String index() {  
    return "index";  
}  
  
@PostMapping(path = "/country") // Map ONLY POST Requests  
public ResponseEntity<String> addNewCountry(@Valid @RequestBody Country country) {  
    // @ResponseBody means the returned String is the response, not a view name  
    // @RequestParam means it is a parameter from the GET or POST request  
    Country c = new Country();  
    c.setName(country.getName());  
    c.setCode(country.getCode());  
    countryRepository.save(c);  
    return ResponseEntity.ok("Country added");  
}  
  
@GetMapping(path = "/country/{id}")  
public ResponseEntity<Country> getOneCountry(@PathVariable Integer id) {  
    Optional<Country> pCountry = countryRepository.findById(id);  
  
    if (pCountry.isEmpty()) {  
        return new ResponseEntity<>(new Country("ERROR", "Impossible d'obtenir le pays portant l'identifiant " + id),  
            HttpStatus.NOT_FOUND);  
    }  
    return new ResponseEntity<>(pCountry.get(), HttpStatus.FOUND);  
}  
  
@GetMapping(path = "/countries")  
public @ResponseBody Iterable<Country> getAllCountries() {  
    // This returns a JSON or XML with the users  
    return countryRepository.findAll();  
}  
  
@PutMapping(path = "/country/{id}")  
public Country updateCountry(@RequestBody Country country) {  
    Country existingCountry = new Country();  
    existingCountry = countryRepository.findById(country.getId()).orElse(null);  
    existingCountry.setName(country.getName());  
    existingCountry.setCode(country.getCode());  
    return countryRepository.save(existingCountry);  
}  
  
@DeleteMapping(path = "/country/{id}")  
public @ResponseBody String deleteCountry(@PathVariable int id) {  
    countryRepository.deleteById(id);  
    return "Country removed !! " + id;  
}
```

DOSSIER PROFESSIONNEL (DP)

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fetch Country </title>
  <link rel="stylesheet" type="text/css" href="styles/countries.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQV3Iipema34MD+dH/1fQ784/j6cY/I3TQUhCwrf7x99vRx72H2w11" cr
</head>
<body>
  <div id="countries" class="m-1">
    <div id="addCountry">
      <label for="name">Country :</label>
      <input type="text" id="name" name="name">
      <label for="code">Code :</label>
      <input type="text" id="code" name="code" minlength="2" maxlength="2">
      <input type="button" value="Envoyer" id="submit">
    </div>
  </div>
</div>
<div class="container">
  <div class="displayCountries d-flex flex-row justify-content-around">
  </div>
</div>
<script type="text/javascript" src="/js/app.js">
</script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41J3QIqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3ZDZwRRNqkq4sF86dIHNDz8K1" crossorigin="anon
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-3jsmVgdy0p3pXb1r1bZUAy011y6OrQ6Vrj1EaFf/nJGzixFdsf4x8xIM+887JRM" crossorigin="anon
</body>
</html>
```

```
main > resources > application.properties
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/restonsconnecter
spring.datasource.username=spring
spring.datasource.password=spring
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
#spring.jpa.show-sql: true
```


DOSSIER PROFESSIONNEL (DP)

```
let displayCountries = document.querySelector('.displayCountries');

fetch('http://localhost:8888/countries',{
  method: 'GET',
  mode: 'no-cors',
  headers: {
    'Content-Type': 'application/json',
  },
}).then(response => response.json())
.then(renderCountries);

function renderCountries(countries) {
  countries.forEach(function (countries,i) {
    country_div = document.createElement('div');
    country_div.id = 'country-div' + i;
    country_div.class = 'country-card';

    country_div.innerHTML = `<div class="card border-0 shadow">
      <div class="card-body text-center">
        <h5 class="card-title mb-0 country-name" id="name-country">${countries.name}</h5>
        <input type="text" class="text-center displayNone country-up-name" id="updateName" value=" ${countries.name}">
        <input type="text" class="text-center displayNone country-up-code" id="updateCode" value="${countries.code}">
        <div class="card-text text-black-50 country-code" id="code-country">
          ${countries.code}</div>
        <a class="btn btn-info see text-small m-2 upbtn" onclick="showUpForm(${i})">
          <small> Modifier</small></a><a class="btn btn-success see text-small m-2 validate displayNone" onclick="updateCountry(${i},${countries.id})">
          <small> Valider</small></a><a class="btn btn-danger delete displayNone small m-2 " onClick="deleteCountry(${countries.id});">
          <small> Supprimer</small>
        </a></div>
      </div>
    `;
    displayCountries.appendChild(country_div);
  })
}

const el = document.getElementById("submit");
el.addEventListener("click", addCountry);
```

DOSSIER PROFESSIONNEL (DP)

```
//Delete
function deleteCountry(id){
  console.log(id)
  let requestOptions = {
    method: 'DELETE',
    redirect: 'follow'
  };
  fetch('http://localhost:8080/country/${id}', requestOptions)
  .then(response => response.json()).then( location.reload())
}
//Put
function updateCountry(i,id){
  updateName = document.querySelectorAll('.country-up-name');
  updateCode = document.querySelectorAll('.country-up-code');
  console.log(updateName[i].value)
  console.log(updateCode[i].value)
  var dataObject = {
    name: updateName[i].value,
    code: updateCode[i].value,
  };
  fetch('http://localhost:8080/country/${id}',{
    method:'PUT',
    headers:{
      'Content-Type':'application/json'
    },
    body:JSON.stringify(dataObject)
  }).then(response=>{
    return response.json()
  }).then( location.reload()).then(data=>
  // this is the data we get after putting our data,
  console.log(data)
  );
}
```

DOSSIER PROFESSIONNEL (DP)

```
function addCountry() {  
  let name = document.getElementById("name");  
  let code = document.getElementById("code");  
  let myHeaders = new Headers();  
  myHeaders.append("Content-Type", "application/json");  
  let raw = JSON.stringify({  
    "name": name.value,  
    "code": code.value  
  });  
  let requestOptions = {  
    method: 'POST',  
    headers: myHeaders,  
    body: raw,  
    redirect: 'follow'  
  };  
  fetch("http://localhost:8080/country", requestOptions)  
    .then(response => response.text())  
    .then(result => console.log(result))  
    .then(location.reload())  
    .catch(error => console.log('error', error));  
}
```

Country : Code : Envoyer

fg
FG

Modifier

Valider

Supprimer

za
AZ

Modifier

Valider

Supprimer

Activité-type 3

Exemple n°1 ▶

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Ca gratte

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ca gratte :

Le projet était fait dans le cas de brief avec simplon.co, pour valider des compétences comme:

- Mettre en place une base de données.
- Développer des composants métier.
- Préparer et exécuter les plans de tests d'une application.
- Préparer et exécuter le déploiement d'une application

Le projet est une application React permettant de créer le jeu du morpion et de voir votre évolution de score. On a créé une fonction history pour mettre en localStorage l'historique des jeux. Ensuite créer le tableau et les boutons pour reset le jeu. La fonction calculate winner pour voir si on a gagné ou pas, si on a gagné on affiche un message pour dire qui est le gagnant.

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Technologie utilisée: Npm. Gitlab, Javascript pour le mécanisme du jeu. Css pour le style.

3. Avec qui avez-vous travaillé ?

J'ai travailler avec Allan Gagne

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service	▶ Ca gratte
Période d'exercice	▶ Du 04.07..2022 au 08.07.2022

5. Informations complémentaires (facultatif)

```
import "./History.css";

function History() {
  const history = JSON.parse(localStorage.getItem("history"))
    ? JSON.parse(localStorage.getItem("history"))
    : [];
  const historyArray = history.reverse();

  return (
    <div>
      <h3>Last Games History</h3>
      <table>
        <tbody>
          {historyArray.map((item, key) => (
            <tr key={key}>
              <td>{item}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default History;
```

DOSSIER PROFESSIONNEL (DP)

```
import './App.css';
import Game from "../components/Game/Game";
import React from "react";

function App() {
  return (
    <div className="App">
      <Game />
    </div>
  );
}

export default App;
```

DOSSIER PROFESSIONNEL (DP)

```
import {fireEvent, render, screen} from '@testing-library/react';
import App from './App';

test('click on board', () => {
  render(<App />);
  const linkElement = screen.getAllByRole('button')
  fireEvent.click(linkElement[0])
  expect(linkElement[0].innerHTML).toEqual('X');
  expect(screen.getByText(/X/g)).toBeInTheDocument();
});

test('click on reset', () => {
  render(<App />);
  const linkElement = screen.getByText('Reset')
  const board = screen.getAllByRole('button', {name:"square"})
  fireEvent.click(linkElement)
  board.forEach((item, i) => {
    expect(board[i].innerHTML).toEqual("");
  })
});
```


DOSSIER PROFESSIONNEL (DP)

```
{
  "name": "morpion",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

DOSSIER PROFESSIONNEL (DP)

```
export function calculateWinner(squares) {  
  const lines = [  
    [0, 1, 2],  
    [3, 4, 5],  
    [6, 7, 8],  
    [0, 3, 6],  
    [1, 4, 7],  
    [2, 5, 8],  
    [0, 4, 8],  
    [2, 4, 6],  
  ];  
  for (let i = 0; i < lines.length; i++) {  
    const [a, b, c] = lines[i];  
  
    if (squares[a] && squares[a] === squares[b] && squares[a] === squares[c]) {  
      return squares[a];  
    }  
  }  
  return null;  
}
```

DOSSIER PROFESSIONNEL (DP)

```
image: node:latest

stages:
  - npm
  - test
npm:
  stage: npm
  script:
    - npm install
  artifacts:
    expire_in: 1 days
    when: on_success
    paths:
      - node_modules/
  rules:
    - when: always

test:
  stage: test
  dependencies:
    - npm
  script:
    - npm test
  rules:
    - when: always
```

Activité-type 3

Exemple n°2 ▶

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Flask test

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Flask test :

Le projet était fait dans le cas de brief avec simplon.co, pour valider des compétences comme:

- Préparer et exécuter les plans de tests d'une application.
- Préparer et exécuter le déploiement d'une application

Le projet est une application permettant de faire du test avec le framework Flask.

Les types des tests .

- Mettre en place un mock pour simuler le retour de fonction
- Mettre au moins 1 "fixture" selon les préconisations de pytest
- Vérifier l'existence de l'app Flask dans un test (par exemple tester le code HTTP)
- Vérification Les paramètres GET des urls si nécessaire avec "parametrize"
- Utiliser les multiple assert
- Catégoriser les résultats pour les exécuter par groupe
- Gérer les exceptions (cas non passant)
- Exporter les résultats en HTML
- Utiliser au moins 1 plugin pytest

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Technologie utilisée: Flask. Gitlab, pipenv.

3. Avec qui avez-vous travaillé ?

J'ai travailler avec Allan Gagne

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service	▶	Flask Test
Période d'exercice	▶	Du 07.04..2022 au 15.04.2022

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

```
from flask import Flask, request
import requests

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/other")
def hello_other():
    page = request.args.get('page', default = 1, type = int)
    return f"<p>Hello, Other!</p><p>Page : {page}</p>"
```

DOSSIER PROFESSIONNEL (DP)

```
@pytest.mark.pageNotFound
@pytest.mark.parametrize("route", ["/james", "/other2"])
def test_page_not_exists(client: Flask, route):
    response = client.test_client().get(str(route))
    assert response.status_code == 404

@pytest.mark.pageBiggerThan500
@pytest.mark.parametrize("page", [501, 600, 982])
def test_page_number_bigger(client, page: int):
    arg = str(page)
    response = client.test_client().get("/other?page=" + arg)
    assert response.status_code == 200
    assert page >= 500

@pytest.mark.pageBiggerThan500Error
@pytest.mark.parametrize("page", [333, 233])
def test_page_number_bigger(client, page: int):
    with pytest.raises(ValueError):
        if page < 500:
            raise ValueError("value must be biggerthan 500")
```

DOSSIER PROFESSIONNEL (DP)

```
@pytest.mark.Success
def test_hello(client: Flask):
    response = client.test_client().get("/")
    assert b"Hello, World" in response.data

def hello():
    return "<p>Hello, James!</p>"

@pytest.mark.Mock
def test_mocked_api(mock):
    mock.patch('test_flask.test_hello', hello )
    assert "Hello, James" in test_hello()

@pytest.mark.pageFound
@pytest.mark.parametrize("page", [2, 4, 7])
def test_content_other_page(client, page: int):
    page = str(page)
    response = client.test_client().get("/other?page=" + page)
    assert response.status_code == 200
```


DOSSIER PROFESSIONNEL (DP)

```
image: python:3.10

stages:
  - pip-job
  - pytest-job

pip-job:
  stage: Install pip and pipenv

  before_script:
    - python -m pip install --upgrade pip
    - pip install pipenv
  script:
    - pipenv install --dev --skip-lock
    - pip install flask
    - pip install requests


pytest-job:
  stage: Install pytest

  before_script:
    - pip install pytest
    - pip install pytest-mock
    - pip install pytest-html

  script:
    - pytest

only:
  - ura_mersad # or the name of your main branch
```

Exporte les résultats en Html.

 report.html	Mersad ura	4 months ago
-------------------------------------------------------------------------------------------------	------------	--------------

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Développeur web et web mobile	Simplon.co	08/2020-03/2021
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.		Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) Mersad Ura ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je
suis l'auteur(e) des réalisations jointes.

Fait à Clermont-ferrand le 22/08/2022

pour faire valoir ce que de droit.

Signature :



DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)