



LOCQMEN HAMDI

SAE 23

MAXIME REDOUTE

PROJET N°8

AMGHAR MOURAD

Le groupe de projet est chargé de développer une application informatique pour les étudiants en BUT1 R&T qui affiche les températures de leur ville de résidence principale et secondaire. Les informations sur la température sont stockées dans une base de données et peuvent être saisies sous forme de fichiers JSON ou CSV, ou directement dans l'interface de la base de données. L'application doit permettre de visualiser la température actuelle et historique de chaque étudiant, ainsi que la température médiane du groupe d'élèves. Les étudiants doivent concevoir cette application sous forme d'un site Web ou d'une application mobile, et documenter les fonctionnalités de l'outil et former les utilisateurs à son utilisation. La SAÉ peut être réalisée par un groupe de 2 ou 3 étudiants, et comprend différentes sections telles que la mise en place de l'environnement de développement, la réalisation documentée de l'application, et l'organisation d'une session de formation en français et en anglais. Les modalités d'évaluation incluent la fonctionnalité et la dynamique du site, la manipulation de données, la documentation technique, un tutoriel d'installation et d'utilisation, le dépôt du code, la démonstration, la présentation de l'outil utilisé pour le développement et la méthode de validation.

Il s'agit de développer une application Web permettant d'afficher la température des lieux de résidence des étudiants d'une formation universitaire. Les données de température peuvent être saisies soit via un fichier json/csv ou via une interface web/mobile. L'application doit également



permettre de stocker ces données dans une base de données et d'afficher la température individuelle et collective des étudiants. Enfin, le professionnel doit documenter les fonctionnalités de l'outil et former les utilisateurs à son utilisation.

RÉPARTITION:

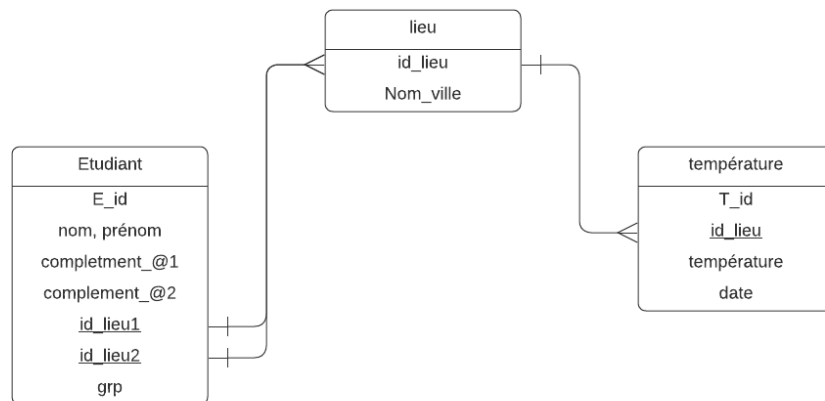
La répartition du travail s'est faite assez rapidement. Locqmen a pris en charge la programmation en Python, car c'était sa spécialité. Maxime s'est occupé de la conception générale de la base du site HTML et du CSS. Quant à Mourad, il a pris en charge le reste du développement du site web, la rédaction du rapport et la préparation de la présentation. En raison de son absence pour des raisons de santé pendant deux jours, Mourad a dû rattraper son retard en travaillant de manière autonome pour terminer son travail dans les délais impartis. En fin de compte, cette répartition du travail a été efficace et a permis à chaque membre de l'équipe de travailler sur les tâches qui correspondaient le mieux à ses compétences et à son expérience, ce qui a permis de produire un travail de qualité dans un temps raisonnable.

CROQUIS:

Pour commencer notre projet, nous avons commencé par faire un croquis de ce que représentait le projet dans son ensemble. Cette étape était cruciale pour comprendre les différentes fonctionnalités nécessaires et ne rien oublier dans la conception et la réalisation du projet.

Diagramme EA de base de données (patte de corbeau)

MaxTM | May 5, 2023



DONNÉES:

Pour commencer notre projet, nous avons fait une base de données sur excel avec le nom , le prénom, l'adresse de résidence principale, celle de la résidence secondaire si il y en a une et du groupe auquel il appartient.

	A	B	C	D	E	F
1	Nom	Prenom	ville 1	ville 2	adresse 1	adresse 2
2	Berrada	Mehdi	Nantes	Montbéliard	15 rue de l'école	100 route d'audincourt
3	chevassu	Matéo	Besançon	Montbéliard	17 rue de la retraite sentimentale	100 route d'audincourt
4	cevik	Altay	Audincourt	//	19 Avenue Jean Jaurès	//
5	redouté	maxime	valentigney	//	25 rue des poirriers	//
6	amghar	mourad	valentigney	//	33 rue de la novie	//
7	benyahia	yanis	audincourt	//	5 rue de la charboniere	//
8	Abouhnaik	Chadi	Montbéliard	Valentigney	3 rue Pierre Donzelot	10 rue Armand peugeot
9	Tolliha	Eric	Belfort	//	5 rue de Moscou	//
0	HARCHAOU	Sami	St loup sur semouse	Montbéliard	41 rue du breuil	23 Rue Mauric Ravel
1	Fouka	Anisse	Audincourt	//	15 rue Leo Ferré	//
2	Zerrar	Yanis	Montbéliard	//	8 rue du buffon	//
3	Hamdi	Locqmen	Audincourt	//	101 Avenue du 8 mai	//
4	Kut	Suha	Besançon	Montbéliard	50B Rue Mirabeau	23 Rue Maurice Ravel
5	Kilic	Fatih	valentigney	//	111 rue des graviers	//
6	nabil	Chaker	Belfort	//	20 Boulevard Anatole France	//
7	Solis Maced	Paul	Le Mesnil amelot	Belfort	1 bis rue de bosnier	1 avenue charles bhon
8	Ockandji	Ardy	Montbéliard	//	37 rue georges clemenceau	//
9	Tchitembo	Richardy	Belfort	//	2 rue Ernest Duvillard	//
0	Nganga	Mario	Montbéliard	//	37 rue georges clemenceau	//
1	ngoma	dorcasse	Montbéliard	//	37 rue georges clemenceau	//
2	Ronaldo	Paulino	Montbéliard	//	37 rue georges clemenceau	//
3	fatih	kurul	montbéliard	mulhouse	24 rue de Audincourt	23 rue de kingersheim
4	nicolas	rabergerau	montebéliard	salins-les-ba	23 maurice ravel	73 rue de la republique
5	wittig	antonin	héricourt	Montbéliard	3 rue des avenues	1 impasse de l'entrée
6	Lucas	antoine	Arbois	Montbéliard	4 rue du château Pécauld	100 route d'Audincourt
7	Prinzbach	Lillian	Meyenheim	Montbéliard	2 impasse des Pommiers	100 route d'Audincourt
8	GAMBIER	Maxence	Montbéliard	//	100 route d'audincourt	//
9	Aoussar	Driss	Grand-charmont	//	4 allée du Hainault	//
0	DOGAN	Eren	Grand-charmont	//	4 rue de Normandie	//
1	MARCHAND	Théo	Triel-sur-Seine	Montbéliard	6 RUE Cadot	36 rue saint Goerges
2	LE GOUSSE	Adrien	Viry-Chatillon	Belfort	14 avenue de l'aqueduc	102 rue de la croix du til
3	Coudert	Yoland	Vichy	Montbéliard	8 avenue de France	8 rue des potiers
4	Thayaparan	Ansuhan	Gromagny	//	38 faubourg de belfort	//
5	guesdon	Mathis	Besancon	Montbéliard	12 impasse le corbusier	25 rue Georges Cuvier
6	DEZZAZ	Yanis	Orléans	Montbéliard	3 rue du jardin dans l'île	23 Rue Maurice Ravel
7	Biermé	Julien	//	Montbéliard	//	23 Rue Maurice Ravel
8	Beaudoux	Timothée	Montbéliard	//	6 Allée du Thiergarten	//
9	Greder	Guillaume	Montbéliard	Vouhenans	23 Rue Maurice Ravel	5 rue de la petite vie
0	Paquelet	Etienne	Bourg-En-Bresse	Montbéliard	250 Chemin des Murailles	100 route d'audincourt
1	Dekhouché	Bilal	Besançon	Montbéliard	22 Rue Danton	23 Rue Maurice Ravel
2	KNOEPFFLER	Xavier	Sermamagny	//	16 Rue de Valdoie	//
3	GALLAY	Dorian	Rang	Mandeure	7 rue de la goulisse	25 rue de Beaulieu

Ensuite, nous l'avons converti en fichier json pour que nous puissions l'utiliser en tant que base de données.

```
HTML PHP+HTML PHP Replace SQL python 2 to 3
1 [{"Nom": "Berrada", "Prenom": "Mehdi", "ville 1": "Nantes", "ville 2": "Montbéliard", "adresse 1": "15 rue de l'école", "adresse 2": "100 route d'audincourt"}, {"Nom": "chevassu", "Prenom": "Mateo", "vil
```



Ensuite, nous avons utilisé l'api "<https://api.open-meteo.com/v1/forecast>" pour avoir les données dynamiques que nous utilisons sur notre site.

Afin que les températures soient toujours mises à jour.

BASE SITE WEB:

HTML:

La base du site web repose sur une seule page HTML dynamique qui ne s'ouvre pas lorsqu'elle est modifiée. De plus, elle permet l'affichage de plusieurs tableaux qui sont remplis avec des données récupérées à l'aide de code Python. Cette approche de conception offre une expérience utilisateur fluide et efficace en réduisant le besoin de recharger les pages et en optimisant la présentation des données. La page HTML dynamique permet à l'utilisateur d'interagir avec le site web et d'accéder à différentes fonctionnalités sans avoir à naviguer à travers plusieurs pages, ce qui le rend plus intuitif et convivial. De plus, le code Python assure que les données affichées sur les tableaux sont à jour et précises, améliorant la fiabilité et l'utilité du site web. Dans l'ensemble, la combinaison d'une page HTML dynamique et de code Python contribue à créer un site web rationalisé et efficace qui offre une expérience utilisateur de haute qualité.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="css/style.css">
<!-- Chargez l'API Google Maps -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjV"
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoVMqMTK8LvdxxYG3nZ448h0EQiglfqkJs1NOQV44ch"
</script>
</head>
<body>
<div id="map">
</div>
<div id="tableau">
</div>
</body>
</html>

function toggleTableau(tableauId) {
    var tableau = document.getElementById(tableauId);
    var tableaux = document.getElementsByTagName('table');
    for (var i = 0; i < tableaux.length; i++) {
        if (tableaux[i].id !== tableauId) {
            tableaux[i].style.display = 'none';
        }
    }
    if (tableau.style.display === 'none') {
        tableau.style.display = 'table';
    } else {
        tableau.style.display = 'none';
    }
}

function showWeatherInfo(adresse, lat, lng, lat2=NaN, lng2=NaN) {
    // Affichez les informations météo
    console.log([lat, lng]);
    var map = L.map('map').setView([lat, lng], 15);
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
        maxZoom: 19
    }).addTo(map);

    // Ajoutez un marqueur à la position de l'adresse
    var marker1 = L.marker([lat, lng]).addTo(map);
    if (lat2!="" && lng2!="") {
        var marker2 = L.marker([lat2, lng2]).addTo(map);
    }
}
```

CSS:

Afin d'améliorer l'aspect visuel de notre page HTML dynamique, nous avons ajouté des styles CSS. Ces styles permettent de modifier l'apparence de la page en appliquant des couleurs, des polices de caractères et des mises en page adaptées à nos besoins. Le CSS permet également de rendre la page plus responsive, c'est-à-dire qu'elle s'adapte automatiquement à la taille de l'écran utilisé, que ce soit sur un ordinateur de bureau ou sur un appareil mobile. En ajoutant du CSS à notre page HTML dynamique, nous avons donc amélioré son apparence et son accessibilité, ce qui contribue à offrir une expérience utilisateur optimale.

```
body {
  background-image: url("../img/fond.jpg");
  background-size: cover;
  font-family: 'Roboto', sans-serif;
}

h1 {
  color: ■aliceblue;
  font-family: 'Montserrat', sans-serif;
  text-align: center;
  font-size: 50px;
}

.wrap {
  text-align: center;
  margin-bottom: 20px;
}

html, body {
  height: 100%;
  text-align: center;
}

.button {
  width: 140px;
  height: 45px;
  font-family: 'Roboto', sans-serif;
  font-size: 11px;
  text-transform: uppercase;
  letter-spacing: 2.5px;
  font-weight: 500;
  color: □#000;
  background-color: ■#fff;
  border: none;
  border-radius: 45px;
}
```

PYTHON:



Le code est une application en Python qui utilise le framework Flask pour créer une application web. L'application effectue les opérations suivantes :

Il importe les modules nécessaires tels que json, Flask, SQLAlchemy, requests et datetime.

Il crée une application Flask et configure la base de données pour stocker les étudiants et leur historique de température.

Il définit une fonction qui nettoie un fichier JSON en supprimant des champs vides. La fonction retourne les données du fichier sous forme de liste Python.

Il définit une fonction qui lit les données d'un fichier JSON et les retourne sous forme de liste Python.

Il définit deux classes de modèles SQLAlchemy, l'une pour les étudiants et l'autre pour l'historique de température. La classe de l'historique de température est liée à la classe de l'étudiant.

Il définit une fonction qui récupère les prévisions météorologiques pour une adresse donnée en utilisant les API Nominatim et Open-Meteo.

Il définit une fonction qui met à jour l'historique de température pour tous les étudiants en récupérant les prévisions météorologiques pour leurs adresses à l'aide de la fonction définie à l'étape 6.

Il planifie l'exécution de la fonction de mise à jour de la température de manière asynchrone à l'aide du module APScheduler.

Il définit des routes Flask pour afficher les étudiants et leur historique de température, ajouter un nouvel étudiant, mettre à jour l'adresse d'un étudiant et supprimer un étudiant.


```
main.py > conditions_meteo

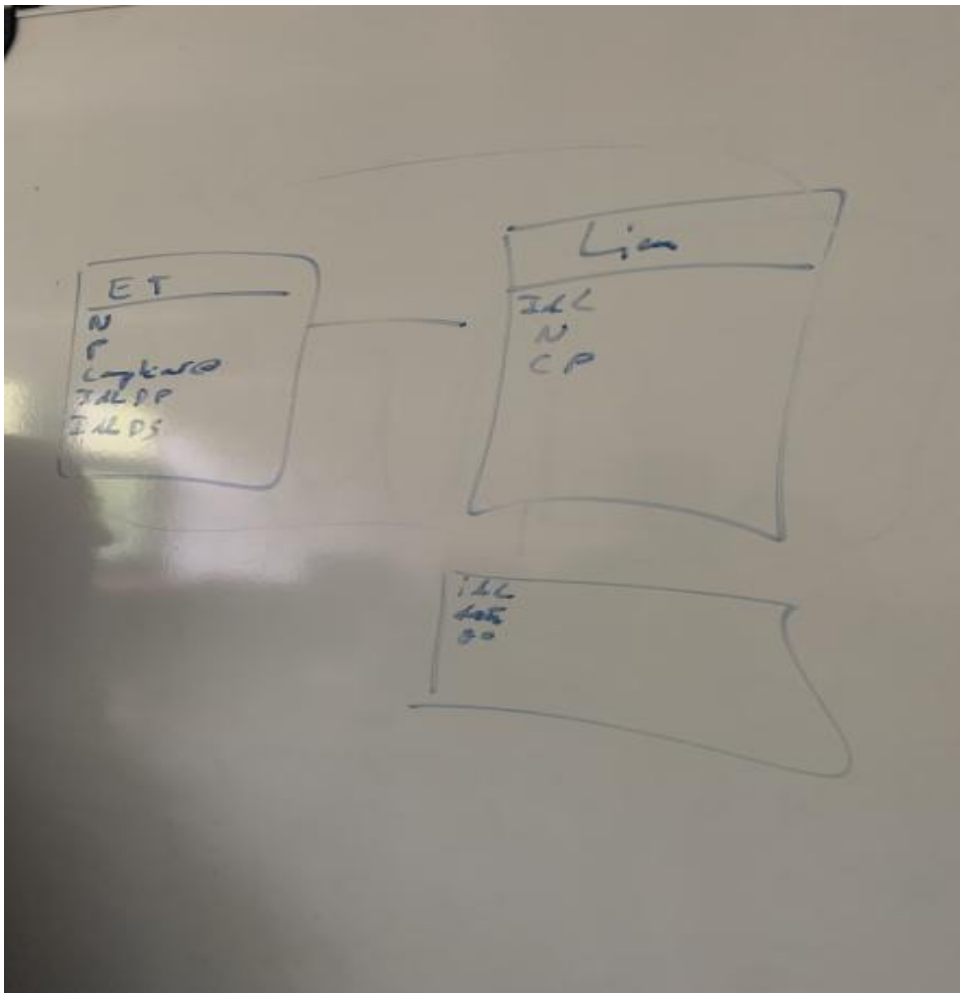
95 def get_meteo(adresse):
96     # Géocodage de l'adresse en utilisant l'API Nominatim d'OpenStreetMap
97     geo_url = "https://nominatim.openstreetmap.org/search"
98     geo_params = {"q": adresse, "format": "json"}
99     geo_response = requests.get(geo_url, params=geo_params)
100
101     if geo_response.status_code != 200:
102         return "Impossible d'obtenir les coordonnées géographiques de l'adresse."
103
104     geo_data = geo_response.json()
105     if not geo_data:
106         return "Adresse introuvable."
107
108     lat = geo_data[0]["lat"]
109     lon = geo_data[0]["lon"]
110
111     # Obtention des prévisions météorologiques à l'aide de l'API Open-Meteo
112     meteo_url = "https://api.open-meteo.com/v1/forecast"
113     meteo_params = {
114         "latitude": lat,
115         "longitude": lon,
116         "current_weather": "true",
117     }
118     meteo_response = requests.get(meteo_url, params=meteo_params)
119
120     if meteo_response.status_code != 200:
121         print(meteo_response.text)
122         return "Impossible d'obtenir les prévisions météorologiques."+meteo_response.text
123
124     meteo_data = meteo_response.json()
125     if not meteo_data or "current_weather" not in meteo_data:
126         return "Météo introuvable."

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL
127.0.0.1 - - [05/May/2023 14:02:46] "GET /?name=amghar%20mourad HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2023 14:02:46] "GET /css/style.css HTTP/1.1" 304 -
```

DIFFICULTÉS:

Lors de la réalisation de notre projet, nous avons rencontré plusieurs difficultés, dont la première et pas des moindres a été de créer un programme qui utilisait seulement deux tables (étudiant et température), alors qu'en réalité

Il fallait utiliser trois tables (étudiant, température et lieux). Cette erreur nous a coûté beaucoup de temps et d'efforts, car elle a entraîné des incohérences dans les données et a rendu difficile l'utilisation du programme. Pour résoudre ce problème, nous avons dû repenser l'architecture de notre base de données et modifier notre code pour tenir compte des trois tables nécessaires. Bien que cela ait été une difficulté importante, cela nous a permis de mieux comprendre l'importance de planifier et de concevoir soigneusement la structure d'une base de données avant de commencer à coder, ce qui nous a évité d'autres problèmes plus tard dans le projet.



La deuxième difficulté que nous avons rencontrée est que le framework Flask que nous avons utilisé pour notre application ne fonctionnait pas sur le



serveur rt-serv. Cela signifie que nous devons trouver une autre solution pour héberger notre application, car le serveur était un élément clé de notre projet. C'est pour cela que nous avons décidé de présenter notre projet en local et non sur le serveur. De plus, nous avons fait un ticket au sein de l'établissement pour que le Flask soit accepté sur le serveur.

En résumé, le groupe de projet constitué de Locqmen, Maxime et Mourad a été chargé de développer une application Web pour les étudiants de BUT1 R&T qui affiche les températures de leur ville de résidence principale et secondaire. Pour ce faire, ils ont développé une base de données en utilisant un fichier JSON pour stocker les informations sur les températures. Ils ont également utilisé l'API "<https://api.open-meteo.com/v1/forecast>" pour avoir les données de température en temps réel. La page web dynamique développée en HTML et CSS offre une expérience utilisateur efficace et intuitive avec des tableaux mis à jour régulièrement grâce au code Python. Chaque membre de l'équipe a travaillé sur les tâches qui correspondaient le mieux à ses compétences et à son expérience, ce qui a permis de produire un travail de qualité dans un temps raisonnable. Enfin, le groupe a travaillé sur la documentation des fonctionnalités de l'outil et a formé les utilisateurs à son utilisation.

VIDÉO DÉMONSTRATION

<https://drive.google.com/drive/folders/1kSS6tr0aNWd3vo8ooy0OZJyinH-WAdZt?usp=sharing>

LIEN PRÉSENTATION

https://www.canva.com/design/DAFh4NPBDJk/J_ouqgS3K_fdsRRph848tA/view?utm_content=DAFh4NPBDJk&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink