

Rapport d'état d'avancement

Compétition ECS'2024

Projet	Projet 12 : Développement d'une plateforme Web immersive via WebXR
Equipe	Oussama Fayz , Youssef Moustaid
Encadrants	M. Mohmed Tabaa , Mme. Houda Mouttalib
Date du rapport	25/07/2024

I. Contexte

Le projet vise à créer une plateforme immersive utilisant des technologies avancées telles que WebXR. Cette plateforme permettra une interaction enrichie dans un environnement 3D pour des fins éducatives.

II. Travaux accomplis durant la semaine

Conception du CMS (Content Management System) :

- Brainstorming des fonctionnalités du CMS.
- Conception du CMS (Content Management System) : Conception du dashboard Admin.
- Finalisation et développement du CMS en utilisant React.js :
 - Création des pages d'administrateur (Étudiants, Ressources, Enseignants, Cours).
 - Création d'un composant dynamique pour la liste de chaque page, incluant les opérations nécessaires avec pagination et fonctionnalité de recherche.
 - Développement d'une page de connexion sécurisée en utilisant des technologies telles que : limitation des tentatives à 3, algorithmes contre le CSS (Cross-Site Scripting).
- Rendre le LMS responsive – compatible avec tous les appareils (Téléphone, Tablette, Ordinateur, etc.).
- Finalisation du développement du LMS : Ajout des fonctionnalités :
 - Confirmation de compte
 - Inscription à un cours
 - Thèmes (sombre ou clair)
 - Mot de passe oublié – Réinitialisation du mot de passe
 - Indicateurs de chargement
 - Course rating.

Développement du Backend du LMS

- Middleware de Sécurité : Ce middleware vérifie les tokens JWT et extrait l'ID et le rôle de l'utilisateur. Il vérifie également que l'ID utilisateur correspond à celui contenu dans le token, empêchant ainsi toute tentative de falsification.
- Gestion des Rôles : Le backend détermine le type d'utilisateur connecté et configure l'affichage du LMS pour qu'il corresponde à ses permissions.
- Une nouvelle fonctionnalité a été ajoutée pour permettre aux utilisateurs (enseignants) de créer des groupes.
- Les enseignants peuvent désormais créer des cours directement depuis l'interface. Cette fonctionnalité inclut la possibilité d'ajouter des descriptions, des dates, et des contenus pédagogiques tels que des images, des fichiers PDF, des modèles 3D, et des vidéos.
- Nous avons implémenté une configuration de multer pour gérer l'upload et le stockage des fichiers de différents formats dans les répertoires appropriés :
 - Images : Stockées dans src/images.
 - Fichiers PDF : Stockés dans src/pdf.
 - Modèles 3D (GLTF, OBJ) : Stockés dans src/objects.

III. Perspectives de travail pour la semaine prochaine

- Début du développement de la plateforme immersive avec l'éditeur de cours.
- Collecte des modèles 3D et avatars nécessaires.

- Démarrage des tests initiaux de la plateforme immersive.
- Récolte des premiers retours et mise en œuvre des améliorations nécessaires.
- Compléter le développement du cms

III. Formations suivies

- Formation en React.js :
 Nom : Create Web Applications with React.js
 Lien : <https://openclassrooms.com/en/courses/7132446-create-a-web-application-with-react-js/7206291-understand-the-purpose-of-react-js>
- Formation en A-frame :
 Nom : Developing AR/VR/MR/XR Apps with WebXR , Unity and Unreal.
 Lien : <https://www.coursera.org/learn/develop-augmented-virtual-mixed-extended-reality-applications-webxr-unity-unreal/home/module/2>
- Formation en Node.js :
 Nom : Developing AR/VR/MR/XR Apps with WebXR , Unity and Unreal.
 Lien : <https://openclassrooms.com/en/courses/5614116-go-full-stack-with-node-js-express-and-mongodb/5614123-set-up-your-coding-environment>

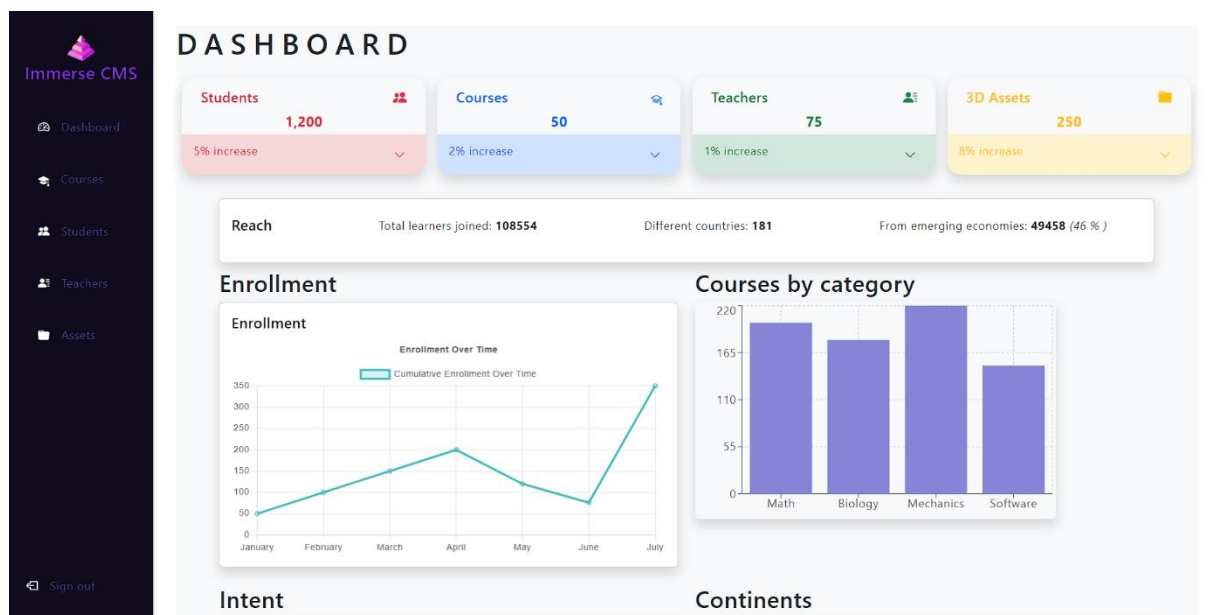



Fig 1 : CMS – Dashboard



Immerse CMS

- Dashboard
- Courses
- Students
- Teachers
- Assets
- Sign out

Courses


Search...

ID	Course Name	Instructor	Credits	Status	Delete	Edit
1	Introduction to Programming	Dr. Alice Brown	3	Ongoing		
2	Advanced Mathematics	Prof. Mark Davis	4	Completed		
3	Modern Physics	Dr. Emily Clark	4	Ongoing		
4	Organic Chemistry	Dr. Robert White	3	Completed		
5	Biology 101	Prof. Laura Green	2	Ongoing		
6	Introduction to Programming	Dr. Alice Brown	3	Ongoing		
7	Advanced Mathematics	Prof. Mark Davis	4	Completed		
8	Modern Physics	Dr. Emily Clark	4	Ongoing		
9	Organic Chemistry	Dr. Robert White	3	Completed		
10	Biology 101	Prof. Laura Green	2	Ongoing		

10

Previous
1
2
3
Next

Fig 2 : Dashboard – Courses



Immerse CMS

- Dashboard
- Courses
- Students
- Teachers
- Assets
- Sign out

Students


Search...

ID	Name	Age	Major	Status	Delete	Edit	Ban
1	John Doe	21	Computer Science	Enrolled			
2	Jane Smith	22	Mathematics	Enrolled			
3	Sam Johnson	20	Physics	Graduated			
4	Emily Davis	23	Biology	Enrolled			
5	Michael Brown	19	Chemistry	Dropped			

10

Previous
1
Next

Fig 3 : CMS – Students


Immerse CMS

- Dashboard
- Courses
- Students
- Teachers
- Assets**
- Sign out

Assets

	Name	Type	Size	Uploaded By	Date	Status	Category	Delete
<input type="checkbox"/>	Model_01.obj	3D Model	15MB	User A	11/07/2023	Active	Architecture	
<input checked="" type="checkbox"/>	Model_02.fbx	3D Model	25MB	User B	12/07/2023	Active	Vehicle	
<input type="checkbox"/>	Texture_01.jpg	Texture	5MB	User C	13/07/2023	Inactive	Materials	
<input checked="" type="checkbox"/>	Animation_01.fbx	Animation	20MB	User D	13/07/2023	Active	Characters	
<input type="checkbox"/>	Model_03.obj	3D Model	30MB	User E	14/07/2023	Inactive	Architecture	
<input checked="" type="checkbox"/>	Model_04.obj	3D Model	10MB	User F	15/07/2023	Active	Environment	
<input type="checkbox"/>	Texture_02.png	Texture	8MB	User G	19/07/2023	Active	Materials	
<input checked="" type="checkbox"/>	Model_05.fbx	3D Model	50MB	User H	20/07/2023	Inactive	Characters	
<input type="checkbox"/>	Animation_02.fbx	Animation	60MB	User I	21/07/2023	Active	Vehicle	
<input checked="" type="checkbox"/>	Model_06.obj	3D Model	40MB	User J	22/07/2023	Active	Furniture	


10 ~

Previous

1

Next

Fig 4 : CMS – Assets












Admin Sign In

Email:


Password:


[Sign In](#)


Fig 5 : CMS – Sign in













Bookmarks





 John Doe
 private Course: Engine Mechanics
 Course
 ★★★★★ (12354)




 JohnDoe
 private Course: Engine Mechanics
 Course
 ★★★★★ (21)




 JohnDoe
 private Course: Engine Mechanics
 Course
 ★★★★★ (21)





 JohnDoe
 private Course: Engine Mechanics
 Course
 ★★★★★ (714)


[Load More](#)


Top Picks for You





 JohnDoe
 public Course: Arduino Basics




 JohnDoe
 private Course: Engine Mechanics












 JohnDoe
 private Course: Engine Mechanics




 JohnDoe
 private Course: Engine Mechanics

All rights reserved © 2024

Fig 6 : LMS – student dashboard



Engine Mechanics
 Course ID: #103

Course Information


Type: private

Category: mechanics

Instructor: John Doe

Description: This is a course to teach basic functionality of mechanical engines in general. We will go over the basics, some components...

Instructor Profile

 John Doe

Rating Insights

4.3 out of 5
 4.3 ★★★★★ 4 ratings

5 star	<div></div>	25%
4 star	<div></div>	25%

All rights reserved © 2024

Fig 7 : LMS – Course information

Immerse

- My Dashboard
- Courses
- Groups
- Students
- Assets
- Profile
- Settings
- Sign Out

Edit Course

Remember, a good course preview image means more traffic for your course



Course ID:
[#103](#)

Course Description

This is a course to teach basic functionality of mechanical engines in general. We will go over the basics, some components...

Save Changes

Course Name

Engine Mechanics

Course Type

Private

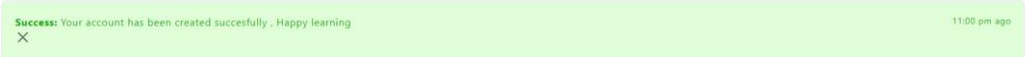
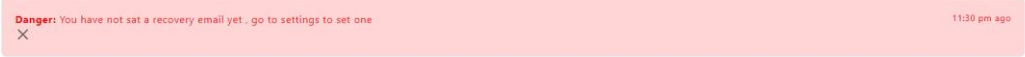

Course Category

Software

All rights reserved © 2024

Fig 8 : LMS – edit course

Search

All rights reserved © 2024

Fig 9 : LMS – Notifications

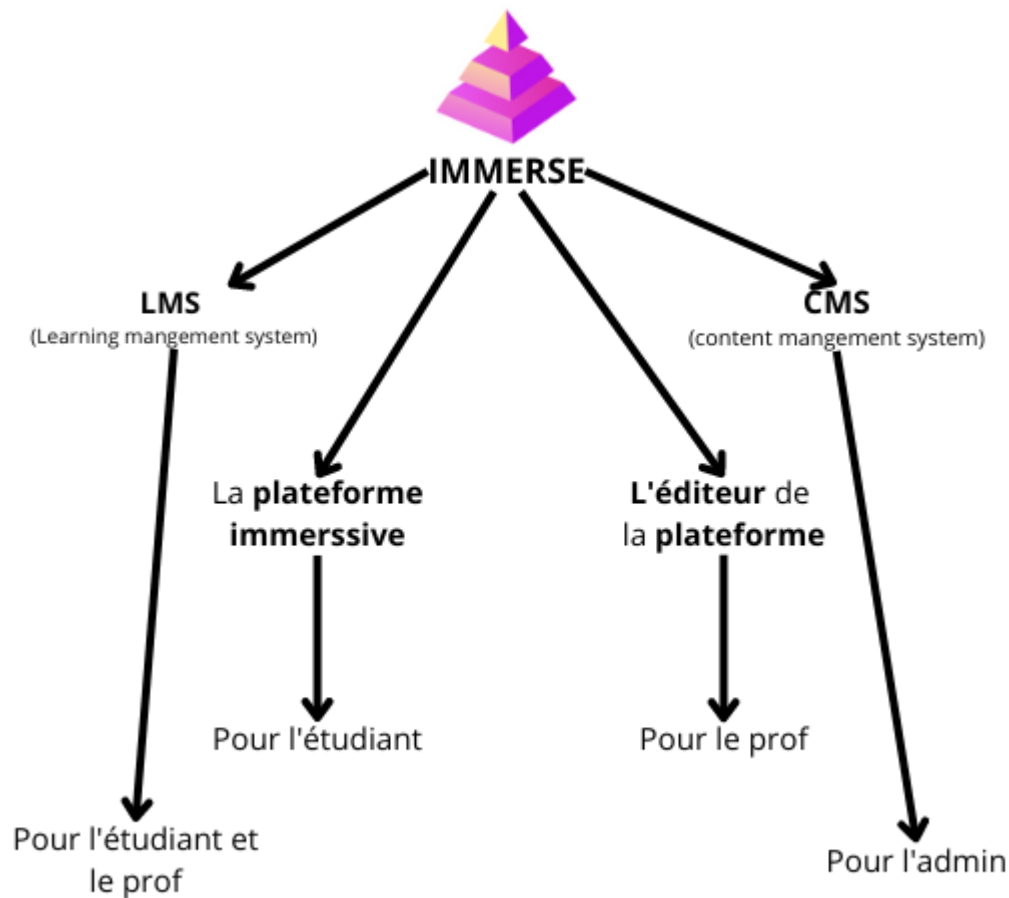


Fig 10: Structure de projet

```

async addCourse() {
  const client = await pool.connect();
  try {
    await client.query('BEGIN');

    // Insert the course into the course table
    const courseResult = await client.query(
      "INSERT INTO course (`idEnvironnement`, `idTeacher`, name, category, `creationDate`, `creationTime`, description, `maxVisitors`, privacy, `coursepd`"
    );
    [this.idEnvironnement, this.idTeacher, this.name, this.category, this.creationDate, this.creationTime, this.description, this.maxVisitors, this.privacy, this.coursepd];

    const courseId = courseResult.rows[0].idCourse;
    // Insert assets into the asset table
    for (let asset of this.assets) {
      await client.query(
        "INSERT INTO asset (`idCourse`, name, type, url, description, size, position, scale, rotation) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9)",
        [courseId, asset.name, asset.type, asset.fileUrl, "None", this.coordinates, this.coordinates, this.coordinates, this.coordinates]
      );
    }
    await client.query('COMMIT');
    console.log("Course and assets added successfully");
  } catch (error) {
    await client.query('ROLLBACK');
    console.error("Error adding course and assets:", error.message);
  } finally {
    client.release();
  }
}

```

Fig 11: Méthode 'addCourse' pour la création d'un cours


```

3 exports.addCourse = async (req, res) => {
4   try {
5     const { idTeacher, title, description, category, environment, privacy, video } = req.body;
6
7     // Initialize file URLs
8     let imageUrl = '';
9     let pdfUrl = '';
10    let additionalAssets = [];
11
12    if (req.files) {
13      const url = req.protocol + '://' + req.get('host');
14
15      // Process all files
16      req.files.forEach(file => {
17        if (file.fieldname === 'previewimage') {
18          imageUrl = url + '/images/' + file.filename;
19        } else if (file.fieldname === 'coursePDF') {
20          pdfUrl = url + '/pdf/' + file.filename;
21        } else if (file.fieldname.startsWith('additionalAsset')) {
22          const index = file.fieldname.split('.')[1];
23          const name = req.body['additionalAsset_' + index + '_name'];
24          const type = req.body['additionalAsset_' + index + '_type'];
25          const fileUrl = url + '/objects/' + file.filename;
26
27          additionalAssets.push({ name, type, fileUrl });
28        }
29      });
30      console.log(additionalAssets);
31    }
32
33    const finalCoursePDF = pdfUrl || '';
34
35    const course = new Course(idTeacher, title, description, category, environment, privacy, finalCoursePDF, video, imageUrl, additionalAssets);
36
37    await course.addCourse();
  }

```

Fig 12: Contrôleur pour la création d'un nouveau cours

```

exports.addGroup = async (req, res) => {
  try {
    const { idTeacher, groupName, description, studentsList } = req.body;

    // Initialize file URLs
    let imageUrl = '';

    // Handle file uploads
    if (req.files) {
      const url = req.protocol + '://' + req.get('host');
      if (req.files.image && req.files.image[0]) {
        imageUrl = url + '/images/' + req.files.image[0].filename;
      }
    }

    const group = new Group(idTeacher, groupName, description, studentsList, imageUrl);
    await group.addGroup();
    res.status(201).json({ message: "Group added successfully" , success: true });
  } catch (error) {
    console.error(error.message);
    res.status(500).json({ error: "An error occurred" , success: false });
  }
};

```

Fig 13: Contrôleur pour la création d'un nouveau groupe

```

api > src > middleware > JS multer-config.js > storage > destination
1  const multer = require('multer');
2  const path = require('path');
3
4  const MIME_TYPES = {
5    'image/jpeg': 'jpg',
6    'image/jpg': 'jpeg',
7    'image/png': 'png',
8    'application/pdf': 'pdf',
9    'audio/mpeg': 'mp3',
10   'audio/wav': 'wav',
11   'model/gltf+json': 'gltf',
12   'application/octet-stream': 'obj',
13 };
14
15 const storage = multer.diskStorage({
16   destination: (req, file, callback) => {
17     console.log("MIME type detected:", file.mimetype);
18     let folder = '';
19     if (file.mimetype.startsWith('image/')) {
20       folder = '../api/src/images';
21     } else if (file.mimetype === 'application/pdf') {
22       folder = '../api/src/pdf';
23     } else if (file.mimetype.startsWith('audio/')) {
24       folder = '../api/src/audio';
25     } else if (file.mimetype.startsWith('model/')) {
26       folder = '../api/src/objects';
27     } else if (file.mimetype.startsWith('application/octet-stream')) {
28       folder = '../api/src/objects';
29     }
30     callback(null, folder);
31   },
32   filename: (req, file, callback) => {
33     const extension = MIME_TYPES[file.mimetype];
34     const name = file.originalname.replace(/s+/g, '_').replace(/\.([^\.]*)$/, '');
35     callback(null, `${name}_${Date.now()}.${extension}`);
36   }
37 });
  
```

Fig 14: Configuration 'multer' pour le téléchargement des différents types de fichiers

```

const jwt = require('jsonwebtoken');

module.exports = (req, res, next) => {
  try {
    const token = req.headers.authorization.split(' ')[1];
    const decodedToken = jwt.verify(token, 'Token_Secret');

    // Extract the user ID and role from the decoded token
    const userId = decodedToken.id;
    const userRole = decodedToken.role;

    // Check for teacher ID or student ID based on the role
    if (
      (userRole === 'teacher' && req.body.teacherid && req.body.teacherid !== userId) ||
      (userRole === 'student' && req.body.studentid && req.body.studentid !== userId)
    ) {
      throw 'Invalid user ID';
    } else {
      next();
    }
  } catch (error) {
    res.status(401).json({
      error: new Error('Invalid request!'),
    });
  }
};
  
```

Fig 15: Vérification de l'identité de l'utilisateur selon son role