

FINPRO

SBD

SISTEM MANAJEMEN

KURSUS ONLINE

GROUP 10

TEAM

SIstem Basis Data



Fadhlureza Sebastian
2306161971



Fauzan Farras Hakim Budi
Handoyo 2306250610

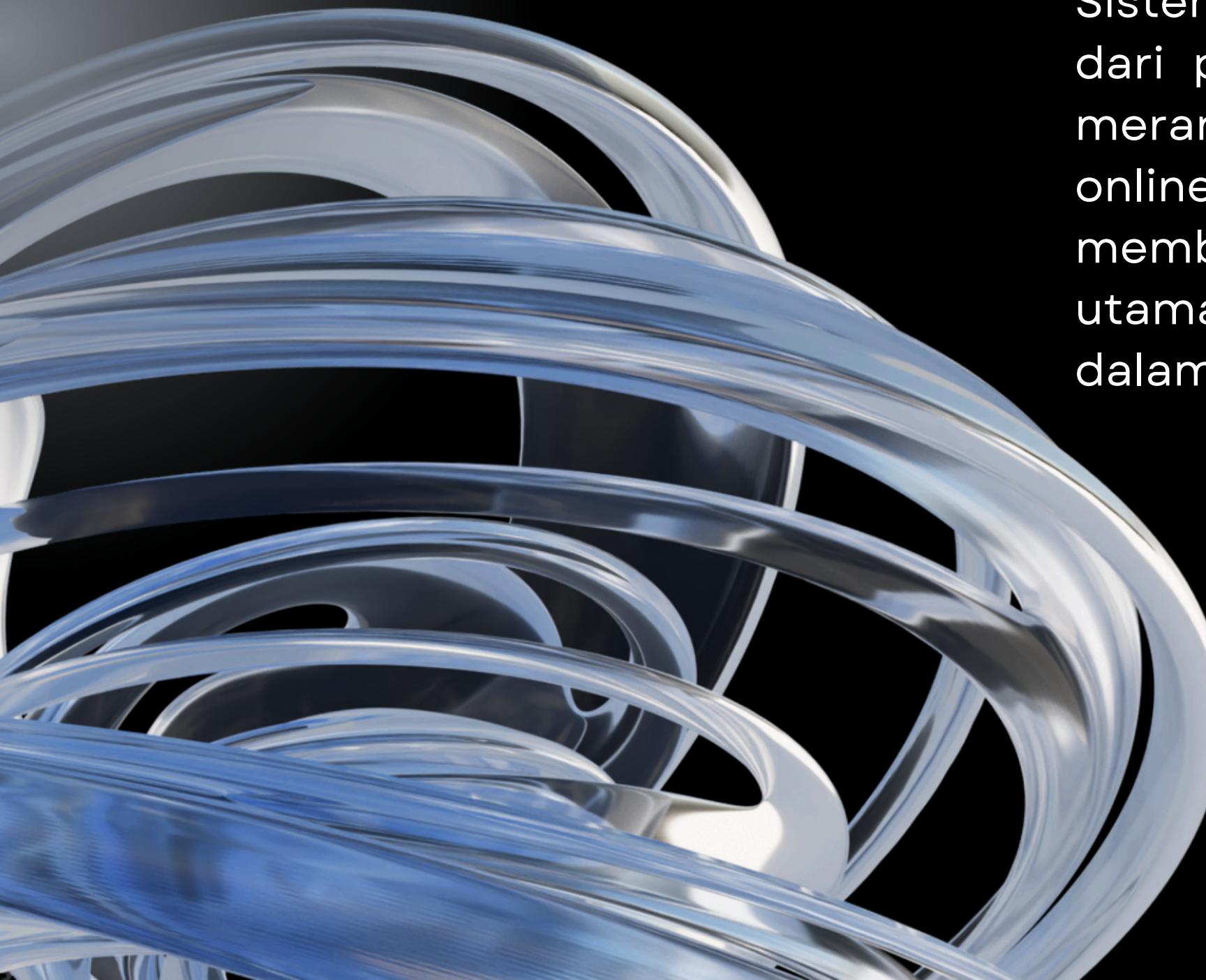


Fido Wahyu Choirulinsan
2306250674



Tri Yoga Arsyad
2306161920

PROJECT BACKGROUND

A large, metallic, reflective spiral shape, possibly made of steel or chrome, is positioned on the left side of the slide. It has a complex, intertwined form that curves and loops across the frame. The lighting highlights the metallic texture and creates sharp shadows against the black background.

Sistem Pengelolaan Kursus Online merupakan tugas akhir dari pelajaran Sistem Basis Data yang bertujuan untuk merancang serta menciptakan aplikasi pembelajaran online yang menarik dan mudah digunakan. Sistem ini memberikan kesempatan kepada dua kategori pengguna utama Pengajar dan Siswa untuk terlibat secara aktif dalam proses pendidikan secara digital.

USER

INSTRUKTUR

Instruktur memiliki hak untuk:

- Mengatur dan mendesain kursus
- Mengupload materi edukasi (dalam format video, PDF, atau PPT)
- Mengubah atau menghapus kursus yang telah dibuat
- Melihat statistik keterlibatan dan perkembangan siswa

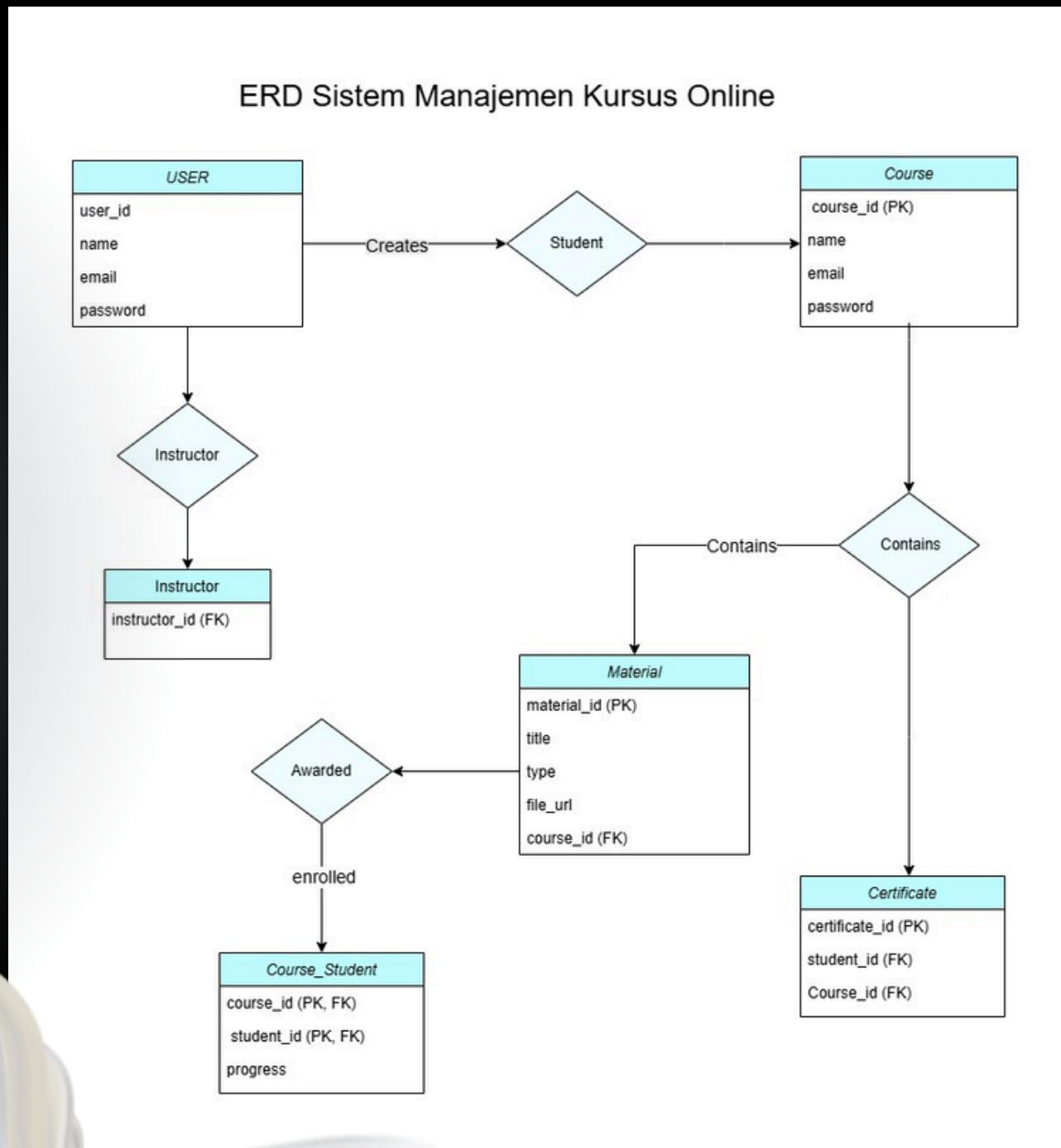
PELAJAR

Pelajar memiliki hak untuk:

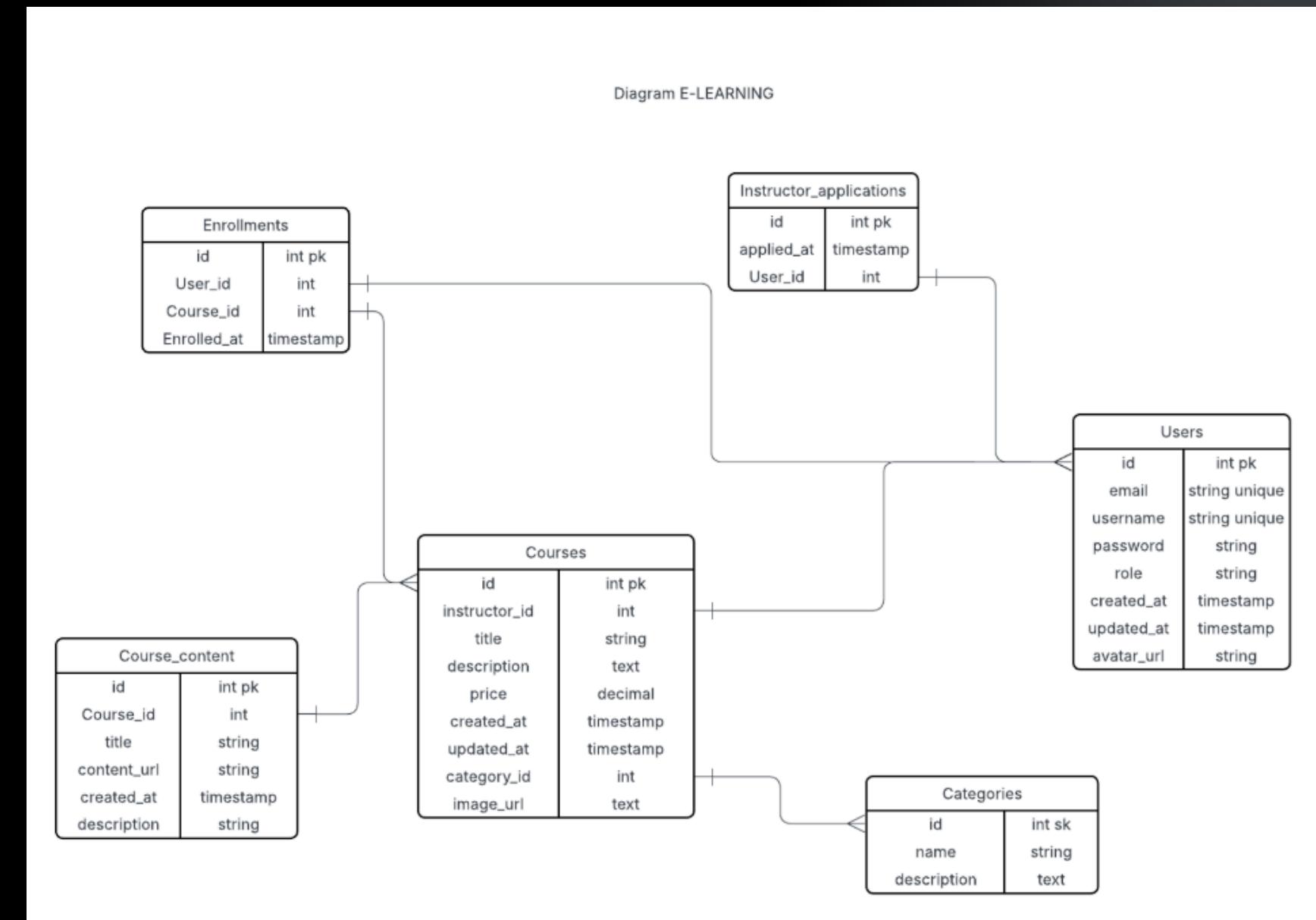
- Mendaftar dan mengikuti kursus yang ada
- Mengakses materi pembelajaran
- Melihat kemajuan belajar pribadi
- Melakukan penilaian dan menerima sertifikat(opsional) setelah menyelesaikan kursus

DIAGRAM

ERD Diagram



UML Diagram



CODE PROGRAM BACKEND

```
1 PORT=5000
2 DATABASE_URL=postgresql://neondb_owner:npg_bVMdfEPKW48r@ep-silent-credit-a1debaaj-pooler.ap-southeast-1.aws.neon.tech/neondb?sslmode=require
3 DATABASE_SSL=false
4 FRONTEND_URL=http://localhost:5173/
5 JWT_SECRET=backend
6
7 CLOUDINARY_CLOUD_NAME=your_cloud_name
8 CLOUDINARY_API_KEY=your_api_key
9 CLOUDINARY_API_SECRET=your_api_secret
```

.env program

File .env menyimpan pengaturan seperti alamat database, nomor port, dan kunci rahasia. Ini juga memuat informasi masuk untuk layanan seperti Cloudinary.



```
E-Learning > backend > db.js > ...
1  require('dotenv').config();
2  const { Pool } = require('pg');
3
4  // Configure the PostgreSQL connection pool
5  const pool = new Pool({
6    connectionString: process.env.DATABASE_URL,
7    ssl: process.env.DATABASE_SSL === 'true' ? { rejectUnauthorized: false } : false, // Disable SSL if not supported
8  });
9
10 module.exports = pool;
```

db.js

File db.js mengelola sambungan ke PostgreSQL dengan memanfaatkan Pool dari pg dan informasi dari .env. Dokumen ini menjamin bahwa koneksi database dapat dimanfaatkan di sektor backend yang lain.

CODE PROGRAM BACKEND

dump.sql

```

1 CREATE TABLE users (
2     id SERIAL PRIMARY KEY,
3     email VARCHAR(255) UNIQUE NOT NULL,
4     username VARCHAR(50) UNIQUE NOT NULL,
5     password VARCHAR(255) NOT NULL,
6     role VARCHAR(20) DEFAULT 'student',
7     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
8     updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
9 );
10
11 CREATE TABLE instructor_applications ( -- for users applying to be instructors
12     id SERIAL PRIMARY KEY, -- Unique identifier for each application
13     user_id INT NOT NULL, -- References the user applying to be an instructor
14     applied_at TIMESTAMP DEFAULT NOW(), -- Timestamp of the application
15     FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
16 );
17
18 CREATE TABLE courses ( -- for courses created by instructors
19     id SERIAL PRIMARY KEY, -- Unique identifier for each course
20     title VARCHAR(255) NOT NULL, -- Title of the course
21     description TEXT, -- Detailed description of the course
22     instructor_id INT NOT NULL, -- References the instructor who created the course
23     price DECIMAL(10, 2) DEFAULT 0.00, -- Price of the course (default is free)
24     created_at TIMESTAMP DEFAULT NOW(), -- Timestamp when the course was created
25     updated_at TIMESTAMP DEFAULT NOW(), -- Timestamp when the course was last updated
26     FOREIGN KEY (instructor_id) REFERENCES users(id) ON DELETE CASCADE
27 );
28
29 CREATE TABLE course_content (
30     id SERIAL PRIMARY KEY,
31     course_id INT NOT NULL, -- References the course the content belongs to
32     title VARCHAR(255) NOT NULL, -- Title of the video content
33     content_url TEXT NOT NULL, -- URL or path to the video file
34     created_at TIMESTAMP DEFAULT NOW(), -- Timestamp when the content was added
35     FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE
36 );
37
38 CREATE TABLE enrollments (
39     id SERIAL PRIMARY KEY, -- Unique identifier for each enrollment
40     user_id INT NOT NULL, -- References the user who enrolled
41     course_id INT NOT NULL, -- References the course the user enrolled in
42     enrolled_at TIMESTAMP DEFAULT NOW(), -- Timestamp when the enrollment occurred
43     FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
44     FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE
45 );
46
47 CREATE TABLE categories (
48     id SERIAL PRIMARY KEY,
49     name VARCHAR(100) NOT NULL,
50     description TEXT
51 );

```

File dump. sql menyusun kerangka tabel untuk basis data seperti pengguna, kursus, dan pendaftaran. pada bagian ini dimanfaatkan untuk memulai atau migrasi awal database pada sistem pembelajaran.

CODE PROGRAM BACKEND

Server.JS

```

1 const express = require('express');
2 const dotenv = require('dotenv');
3 const cors = require('cors');
4 const pool = require('./db'); // Import the centralized db.js file
5 const userRoutes = require('./src/routes/userRoutes'); // Import user routes
6 const authRoutes = require('./src/routes/authRoutes'); // Import auth routes
7 const instructorRoutes = require('./src/routes/instructorRoutes'); // Import instructor routes
8 const courseRoutes = require('./src/routes/coursesRoutes'); // Import course routes
9 const enrollmentRoutes = require('./src/routes/enrollmentRoutes'); // Import enrollment routes
10 const courseContentRoutes = require('./src/routes/courseContentRoutes'); // Import course content routes
11 const categoryRoutes = require('./src/routes/categoryRoutes'); // Import category routes
12
13 // Load environment variables
14 dotenv.config();
15
16 const app = express();
17 const PORT = process.env.PORT || 5000;
18
19 app.use(express.json());
20
21 // Enable CORS
22 app.use(cors({
23   origin: '*', // In production, specify your frontend origin
24   methods: ['GET', 'POST', 'PUT', 'DELETE'],
25   allowedHeaders: ['Content-Type', 'Authorization']
26 }));
27
28 // Test route
29 app.get('/', (req, res) => {
30   res.send(`* Online E-Learning Backend is running!`);
31 });
32
33 // Auth routes
34 app.use('/api/auth', authRoutes); // Mount auth routes
35

```

File server.js mengoperasikan server Express, mengkonfigurasi middleware, dan mengaitkan rute API dengan basis data. Bagian ini berfungsi sebagai pusat utama untuk backend yang mengelola permintaan dari frontend.

```

36 // User routes
37 app.use('/api/users', userRoutes); // Mount user routes
38 app.use('/api/instructors', instructorRoutes); // Mount instructor routes
39 app.use('/api/courses', courseRoutes); // Mount course routes
40 app.use('/api/enrollments', enrollmentRoutes); // Mount enrollment routes
41 app.use('/api/course-contents', courseContentRoutes); // Mount course content routes
42 app.use('/api/categories', categoryRoutes); // Mount category routes
43
44 // Test database connection
45 pool.connect()
46   .then(() => console.log('Connected to PostgreSQL database'))
47   .catch((err) => console.error('Failed to connect to PostgreSQL:', err));
48
49 // Start server
50 app.listen(PORT, () => {
51   console.log(`Server berjalan di http://localhost:\${PORT}`);
52 });

```

CODE PROGRAM FRONTEND

File api.js mengatur hubungan antara frontend dan backend menggunakan axios serta token autentikasi. Dokumen ini juga menawarkan berbagai fungsi API seperti login, register, dan kelola course.

```
31 // Course API calls
32 export const courseAPI = {
33   getAllCourses: () => api.get('/courses'),
34   getCourseById: (id) => api.get(`/courses/${id}`),
35   createCourse: (courseData) => api.post('/courses', courseData),
36   updateCourse: (id, courseData) => api.put(`/courses/${id}`, courseData),
37   deleteCourse: (id) => api.delete(`/courses/${id}`),
38 };
39
40 export const fetchEnrolledCourses = async () => {
41   try {
42     // Use the configured api instance instead of axios directly
43     const response = await api.get('/users/enrolled');
44     return response.data;
45   } catch (error) {
46     console.error('Error fetching enrolled courses:', error);
47     throw error;
48   }
49 };
50
51 export default api;
```

api.JS

```
1 import axios from 'axios';
2
3 // Use environment variable for the base URL
4 const api = axios.create({
5   baseURL: 'http://localhost:5000/api', // Fallback to localhost for development
6   headers: {
7     'Content-Type': 'application/json',
8   },
9 });
10
11 // Add request interceptor to include auth token in requests
12 api.interceptors.request.use(
13   (config) => {
14     const token = localStorage.getItem('token');
15     if (token) {
16       config.headers.Authorization = `Bearer ${token}`;
17     }
18     return config;
19   },
20   (error) => Promise.reject(error)
21 );
22
23 // Auth API calls
24 export const authAPI = {
25   register: (userData) => api.post('/users/register', userData),
26   login: (userData) => api.post('/users/login', userData),
27   logout: () => api.post('/users/logout'),
28   getCurrentUser: () => api.get('/users/me'),
29 };
30
```

CODE PROGRAM FRONTEND

courseService.js

```
1 import api from './api';
2
3 export const courseService = {
4   createCourse: (courseData) => api.post('/courses', courseData),
5   getCourses: () => api.get('/courses'),
6   updateCourse: (id, courseData) => api.put(`/courses/${id}`, courseData),
7   deleteCourse: (id) => api.delete(`/courses/${id}`)
8 };
9
10
11 // Atau jika kita ingin menggunakan named exports.
12 export const createCourse = (courseData) => api.post('/courses', courseData);
13 export const getCourses = () => api.get('/courses');
```

File courseService.js mempermudah akses API course di bagian frontend dengan mengemas fungsi CRUD. Dokumen ini membantu dalam pengelolaan data course di dalam komponen React.

tailwind.config.js

```
1 /**
2  * @type {import('tailwindcss').Config}
3 */
4 export default {
5   content: [
6     './index.html',
7     './src/**/*.{js,ts,jsx,tsx}',
8   ],
9   theme: {
10     extend: {
11       colors: {
12         primary: '#2563eb',
13         secondary: '#1e40af',
14       },
15     },
16   },
17 }
```

File tailwind.config.js mengatur pengaturan Tailwind seperti jalur file yang digunakan dan penyesuaian tema. Dokumen ini memungkinkan penyesuaian warna dan gaya sesuai dengan kebutuhan proyek.

CODE PROGRAM FRONTEND

App.jsx

```

1 import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';
2 import { AuthProvider } from './context/AuthContext';
3 import Navbar from './components/Navbar/Navbar'; // Pastikan path sesuai struktur folder
4 import Home from './pages/Home';
5 import Login from './pages/Login';
6 import Register from './pages/Register';
7 import StudentDashboard from './pages/StudentDashboard';
8 import InstructorDashboard from './pages/InstructorDashboard';
9 import ProtectedRoute from './components/ProtectedRoute';
10 import NotFound from './pages/NotFound';
11 import EnrolledCourses from './pages/EnrolledCourses'; // New import
12 import UserProfile from './pages/ProfilePage';
13 import InstructorCourses from './pages/instructor/InstructorCourses';
14 import CreateCourse from './pages/instructor/CreateCourse';
15 import EditCourse from './pages/instructor/EditCourse';
16
17 export default function App() {
18   return (
19     <Router>
20       <AuthProvider>
21         <Navbar />
22         <main className="container mx-auto p-4 mt-16"> /* Tambahkan mt-16 untuk offset navbar fixed */
23           <Routes>
24             {/* Public Routes */}
25             <Route path="/" element={<Home />} />
26             <Route path="/login" element={<Login />} />
27             <Route path="/register" element={<Register />} />
28
29             {/* Protected Routes for all authenticated users */}
30             <Route element={<ProtectedRoute allowedRoles={['student', 'instructor']} />}>
31               <Route path="/profile" element={<UserProfile />} />
32               <Route path="/main" element={<StudentDashboard />} /> /* Renamed from '/student' to '/main' */
33               <Route path="/enrolled" element={<EnrolledCourses />} />
34             </Route>
35
36             {/* Keep the original '/student' route for backward compatibility */}
37             <Route element={<ProtectedRoute allowedRoles={['student', 'instructor']} />}>
38               <Route path="/student" element={<Navigate to="/main" replace />} />
39             </Route>
40
41             {/* Redirect instructor dashboard to main */}
42             <Route element={<ProtectedRoute allowedRoles={['instructor']} />}>
43               <Route path="/instructor" element={<Navigate to="/main" replace />} />
44               <Route path="/instructor/courses" element={<InstructorCourses />} />
45               <Route path="/instructor/create-course" element={<CreateCourse />} />
46               <Route path="/instructor/courses/:courseId/edit" element={<EditCourse />} />
47             </Route>
48
49             {/* 404 Not Found */}
50             <Route path="*" element={<NotFound />} />
51           </Routes>
52         </main>
53       </AuthProvider>
54     </Router>
55   );
56 }

```

File App.jsx berfungsi guna untuk mengatur jalur dalam aplikasi dengan memanfaatkan React Router dan mengelola kontrol akses sesuai dengan peran pengguna. Ini menghubungkan halaman-halaman seperti masuk, dasbor, dan kursus dengan sistem navigasi dan otorisasi.

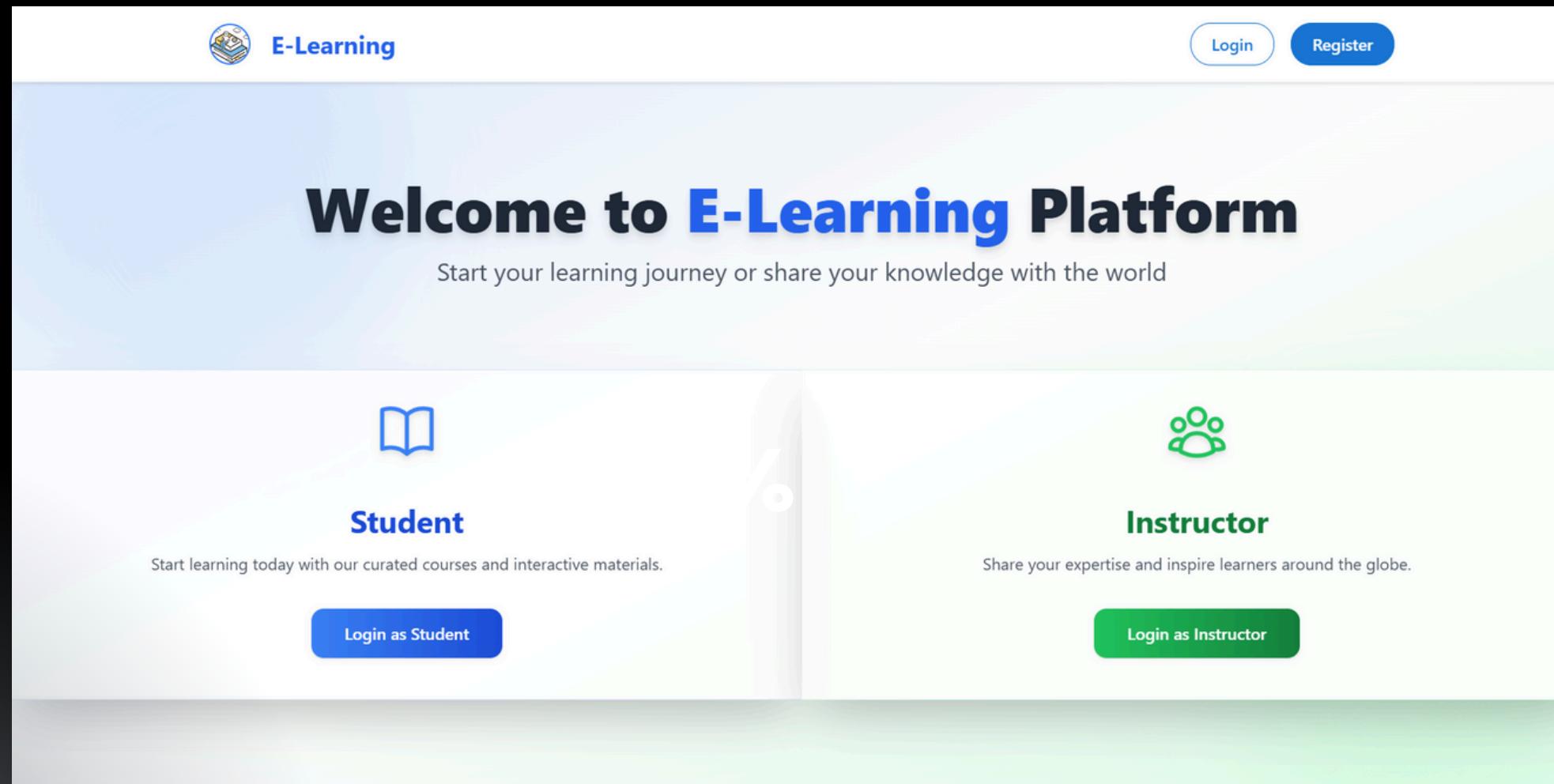
WEBSITE DOCUMENTATION

SISTEM MANAJEMEN KURSUS ONLINE

Platform ini adalah aplikasi berbasis web (yang juga dapat diadaptasi untuk versi mobile) yang menawarkan fitur-fitur utama untuk mendukung kegiatan belajar secara online.

[LINK AKSES REPO GITHUB FINPRO SBD10](#)

Tampilan Awal Website



Tampilan awal ini merupakan halaman pertama yang terlihat oleh pengguna ketika mengakses situs web, baik calon pengajar maupun pelajar. Di sini, pengguna bisa dengan gampang memilih untuk mendaftar sebagai pelajar atau pengajar. Desainnya dibuat sederhana agar proses pendaftaran dan masuk lebih mudah, memberikan pengalaman pertama yang menyenangkan dan efektif bagi seluruh pengguna.

Registration

The registration page features a header with the 'E-Learning' logo, 'Login', and 'Register' buttons. To the right is a green sidebar with a clipboard icon and the text 'Create Account'. It encourages users to 'Join our platform and start your learning journey'. Below this are 'Sign up with Google' and 'or continue with email' options. The main form includes fields for 'Username' (yoga12@gmail.com), 'Email Address' (your@email.com), 'Password' (a placeholder '...'), and 'I am a' (a dropdown menu). A note states 'Password must be at least 8 characters long'.

Menu Login

The login page has a header with the 'E-Learning' logo, 'Login', and 'Register' buttons. On the left, there's a blue sidebar with a 'Join Community' button and a 'Connect with other learners and share your knowledge.' message. The main area features a large 'Log In' button with a circular arrow icon. Below it is a field for 'Email Address' (yoga12@gmail.com) and a 'Forgot password?' link. The 'Password' field contains a placeholder '...'. A 'Log in to your account' button is at the bottom, followed by a 'Create account' link. The background has three small circular icons.

Halaman masuk ini adalah antarmuka yang ditampilkan ketika pengguna memilih untuk mengakses sistem setelah mereka mendaftar. Penampilannya dirancang agar menarik dan mudah digunakan, untuk memberikan kesan yang baik serta dorongan saat memulai pembelajaran di platform tersebut.

Halaman Registrasi digunakan oleh pengguna baik pengajar maupun siswa untuk membuat akun sebelum bisa mengakses platform pembelajaran. Pengguna diminta untuk mengisi informasi sesuai dengan peran yang mereka miliki. Desain halaman ini dibuat menarik dan mudah dipahami, agar meningkatkan semangat pengguna untuk memulai proses belajar secara online.

Course View

The screenshot shows the 'Enrolled Courses' page. At the top, there is a header with the logo, 'E-Learning', and navigation links: 'Welcome, Fadhlureza Sebastian', 'Dashboard', 'Enrolled Courses', 'Profile', and a red 'Logout' button. Below the header, the title 'My Enrolled Courses' is displayed. Three course cards are listed:

- Learn Advanced Python**: An advanced Python course for experienced developers. By: Instructor. Continue Learning button.
- Learn Advanced Python**: An advanced Python course for experienced developers. By: Instructor. Continue Learning button.
- Hello**: sdadsdsdaaa. By: Instructor. Continue Learning button.

Halaman Enrolled Courses menampilkan daftar kursus yang telah diikuti oleh pengguna. Pada halaman ini, setiap kursus ditampilkan dalam bentuk card yang berisi informasi penting seperti judul kursus dan progress pembelajaran. Jika pengguna belum mengikuti kursus apapun, sistem akan menampilkan pesan "You haven't enrolled in any courses yet" beserta tombol untuk menjelajahi kursus yang tersedia.

The screenshot shows the 'Course View' page. At the top, there is a header with the logo, 'E-Learning', and navigation links: 'Welcome, Fadhlureza Sebastian', 'Dashboard', 'Enrolled Courses', 'Profile', and a red 'Logout' button. Below the header, there is a sidebar with the text '← Back to Dashboard' and 'sss'. The main content area displays a card titled 'Course Content' with the text 'Hello'. A blue button labeled 'Open in New Tab' is located in the top right corner of the card.

Halaman Course View merupakan tampilan utama saat pengguna mengakses materi pembelajaran yang telah diikuti. Pada bagian kiri terdapat Course Content yang menampilkan daftar materi pembelajaran dalam bentuk sidebar terstruktur. Di bagian kanan, materi yang dipilih akan ditampilkan dalam format PDF. Pengguna juga diberikan opsi "Open in New Tab" untuk membuka materi dalam tab baru demi kenyamanan dalam belajar. Desain halaman dibuat modern dengan latar putih dan efek bayangan yang memberikan kesan profesional serta mudah digunakan.

PROJECT CONCLUSION

- **Platform Terpadu:** Sistem pembelajaran online yang kami rancang untuk pengajar dan siswa dengan antarmuka yang responsif untuk akses di web dan perangkat mobile.
- **Fitur Berdasarkan Peran:** Instruktur dapat merancang dan mengelola kelas, sementara siswa dapat belajar dan menerima sertifikat (opsional).
- **Teknologi Modern:** Memanfaatkan Express.js, React, PostgreSQL, dan JWT untuk meningkatkan kinerja dan menjaga keamanan.
- **Manajemen Data Rapi:** Database yang terstruktur dengan baik dilengkapi relasi yang jelas dan Redis untuk mempercepat akses data.
- **Dukungan Pendidikan Digital:** Memfasilitasi pembelajaran yang fleksibel dan menunjukkan integrasi dengan database nyata.



THANK YOU

TECH COMPANY