**MFUMO WA USIMAMIZI WA MITAALA (CURRICULUM MANAGEMENT SYSTEM)**

**MUHTASARI WA MFUMO**

Huu ni mfumo wa kisasa wa usimamizi wa mitaala wa chuo kikuu unaotumia teknolojia za hali ya juu za web development. Mfumo umeundwa kwa kutumia **Frontend** na **Backend** zilizounganishwa kwa njia ya kitaalamu.

**MUUNDO WA KIMSINGI (SYSTEM ARCHITECTURE)**

1. **FRONTEND (UPANDE WA MTUMIAJI)**

**Teknolojia Zilizotumika:**

**React 18:** - JavaScript library ya kujenga user interfaces

**TypeScript** :- Lugha ya programming yenye type safety

**Vite**:- Build tool ya haraka ya modern web development

**Tailwind CSS**:- Utility-first CSS framework

**Jinsi Inavyofanya Kazi:**

- Inaendesha kwenye **Port 5173** (http://localhost:5173)

- Inatumia **Single Page Application (SPA)** architecture

- **React Router** inatumika kwa navigation

- **Context API** inatumika kwa state management

- **Hooks** (useState, useEffect) zinatumika kwa data management

**2. BACKEND (UPANDE WA SERVER)**

**Teknolojia Zilizotumika:**

- **Node.js** - JavaScript runtime environment

- **Express.js** - Web framework ya Node.js

- **TypeScript** - Type-safe programming

- **PostgreSQL** - Relational database

- **CORS** - Cross-Origin Resource Sharing

-**Dotenv** - Environment variables management

**Jinsi Inavyofanya Kazi:**

- Inaendesha kwenye **Port 3001** (http://localhost:3001)

- Inatoa **RESTful API endpoints**

- Inaunganishwa na **PostgreSQL database**

- Inatumia **connection pooling** kwa database efficiency

**MUUNGANISHO WA FRONTEND NA BACKEND**

**1. API COMMUNICATION PATTERN**

**Frontend → Backend Communication:**

```javascript

**// Mfano wa API call kutoka frontend**

const fetchColleges = async () => {

  try {

    const response = await fetch('http://localhost:3001/api/colleges');

    const data = await response.json();

    return data;

  } catch (error) {

    console.error('Error fetching colleges:', error);

  }

};

```

**Backend Response Pattern:**

```javascript

**// Mfano wa API endpoint kwenye backend**

app.get('/api/colleges', async (req, res) => {

  try {

    const result = await pool.query('SELECT \* FROM colleges');

    res.json(result.rows);

  } catch (error) {

    res.status(500).json({ error: 'Database error' });

  }

});

```

**### 2. DATA FLOW (MTIRIRIKO WA DATA)**

```

[User Interface] → [Frontend React] → [HTTP Request] → [Backend Express] → [PostgreSQL Database]

                                    ←                ←                  ←

[UI Updates]     ← [State Update]   ← [JSON Response] ← [Database Query] ← [Data Storage]

```

**DATABASE SCHEMA (MUUNDO WA DATABASE)**

**Jedwali Kuu (Main Tables):**

1. **colleges** - Vyuo vya chuo kikuu

2. **departments** - Idara za kila chuo

3. **programs** - Mipango ya masomo (degrees)

4. **courses** - Masomo binafsi

5. **semesters** - Vipindi vya masomo

6. **curriculums\***- Mitaala kamili

**Uhusiano wa Jedwali (Table Relationships):**

```

colleges (1) → (many) departments

departments (1) → (many) programs

programs (1) → (many) courses

semesters (1) → (many) courses

```

---

**JINSI MFUMO UNAVYOFANYA KAZI (SYSTEM WORKFLOW)**

**1. KUANZA MFUMO (SYSTEM STARTUP)**

```

1. PostgreSQL Database inaanza

2. Backend Server inaanza (Port 3001)

3. Frontend Server inaanza (Port 5173)

4. Browser inafungua http://localhost:5173

```

**2. DATA LOADING PROCESS**

```

1. Frontend inapakia (React components load)

2. DataContext inafanya API calls

3. Backend inapokea requests

4. Database queries zinaendeshwa

5. Data inarudishwa kwa JSON format

6. Frontend inaupdate UI na data mpya

```

**3. USER INTERACTION FLOW**

```

Mtumiaji anabofya button → Frontend inasend request → Backend inaprocess → Database inahifadhi → Response inarudi → UI inaupdate

```

**VIPENGELE VYA MFUMO (SYSTEM COMPONENTS)**

**Frontend Components:**

- **DashboardView** - Dashboard kuu

- **CollegesView** - Usimamizi wa vyuo

- **DepartmentsView** - Usimamizi wa idara

- **ProgramsView** - Usimamizi wa mipango

- **CoursesView** - Usimamizi wa masomo

- **CurriculumView** - Usimamizi wa mitaala

**Backend Routes:**

- **/api/colleges** - CRUD operations za vyuo

- **/api/departments** - CRUD operations za idara

- **/api/programs**- CRUD operations za mipango

- **/api/courses** - CRUD operations za masomo

- **/api/curriculums** - CRUD operations za mitaala

**MAZINGIRA YA DEVELOPMENT (DEVELOPMENT ENVIRONMENT)**

**Environment Variables (.env)**

DB\_PORT=54DB\_NAME=curriculum\_db

DB\_USER=postgres

DB\_PASSWORD=your\_password

PORT=3001

```

**Package Management:**

- **Frontend:**npm/yarn packages za React ecosystem

- **Backend:** npm packages za Node.js ecosystem

**SECURITY NA PERFORMANCE**

**Security Measures:**

-**CORS configuration** kwa API security

- **Environment variables**  kwa sensitive data

-**Input validation** kwenye backend

-**Error handling** kwa user experience

**Performance Optimization:**

- **Connection pooling** kwa database

- **Async/await** kwa non-blocking operations

- **Component optimization** kwenye React

- **Efficient database queries**:

---

**FAIDA ZA MUUNDO HUU**

1. **Scalability** - Mfumo unaweza kukua kwa urahisi

2. **Maintainability** - Code ni rahisi kutunza

3. **Separation of Concerns**- Frontend na Backend zimetenganishwa

4. **Modern Technologies** - Inatumia teknolojia za kisasa

5. **Type Safety** - TypeScript inahakikisha code quality

6. **Responsive Design** - Inafanya kazi kwenye vifaa vyote

**(DEVELOPMENT WORKFLOW)**

**Development Process:**

1. **Database Design** → Schema creation

2. **Backend Development**→ API endpoints

3. **Frontend Development** → UI components

4. **Integration** → Connect frontend na backend

5. **Testing** → Ensure everything works

6. **Deployment** → Production ready

**Best Practices:**

- **Code organization** kwa maintainability

- **Error handling** kwa user experience

- **Documentation** kwa team collaboration

- **Version control** kwa code management

*Mfumo huu umeundwa kwa kutumia mbinu za kisasa za software engineering na unafuata international standards za web development. By MUST CURRICULUM DEVELOPMENT GROUP (SOFTWARE DEVELOPMENT 2025)*