

Week-IX Milestone

Environment Setup

For SmartDocQ, it is critical to have separate environment settings to ensure smooth development, testing, and deployment. We have established three main environments:

Development (Local)

Used by developers on their machines.

Allows debugging and testing without affecting live data.

Staging

Acts as a pre-production environment.

Mirrors the production setup closely, used for testing new releases before going live.

Production (Live)

The live version accessed by real users.

Uses secure keys and production-grade database and storage services.

1.1 Environment Variables

Environment variables allow us to **keep sensitive information and configuration separate from the code**. They are stored in files like:

- `.env.development` → For local development
- `.env.staging` → For staging server
- `.env.production` → For production server

Example variables used in SmartDocQ:

General

`NODE_ENV=development` # Current environment type

`PORT=5000` # Port for the server

Database

`MONGO_URI=mongodb://localhost:27017/smartdocq` # Local MongoDB

Authentication

JWT_SECRET=your_secret_key_here # JWT signing key

Cloud Storage

CLOUDINARY_URL=your_cloudinary_url_here # For file uploads

AI Service

GEMINI_API_KEY=your_api_key_here # Key to access the AI Q&A model

Explanation:

NODE_ENV: Lets the app know which environment it is running in. Libraries often use it to enable/disable debug logs or optimize performance.

PORT: The port number your SmartDocQ backend server listens to.

MONGO_URI: Database connection string. Each environment uses its own database to avoid mixing test and live data.

JWT_SECRET: Secret key used to generate JWT tokens for user authentication. Must remain private.

CLOUDINARY_URL: Cloudinary credentials for storing and managing uploaded files.

GEMINI_API_KEY: API key used to connect SmartDocQ with the AI-powered Q&A engine.

Security Note: .env files should never be committed to version control. They must be included in .gitignore to prevent leaking secrets.

1.2 Central Configuration (config.js)

To avoid using process.env everywhere, SmartDocQ uses a central configuration file:

```
import dotenv from "dotenv";
```

```
dotenv.config(); // Loads variables from the appropriate .env file
```

```
export default {
```

```
  env: process.env.NODE_ENV || "development",
```

```
  port: process.env.PORT || 5000,
```

```
mongoUri: process.env.MONGO_URI,  
jwtSecret: process.env.JWT_SECRET,  
cloudinaryUrl: process.env.CLOUDINARY_URL,  
geminiApiKey: process.env.GEMINI_API_KEY,  
};
```

How It Works:

1. dotenv.config() reads the .env file corresponding to the current environment.
2. All configuration variables are exposed via a single config object.
3. Anywhere in SmartDocQ, you can simply import it:

```
import config from './config.js';  
  
console.log(config.port);    // Automatically uses the correct port  
  
console.log(config.mongoUri); // Connects to the right DB for the  
environment
```

2. API Documentation for SmartDocQ

Clear documentation is essential so that anyone (developer, tester, or reviewer) can understand and use the APIs.

2.1 Authentication APIs

POST /api/auth/signup – Register a new user

Request:

```
{  
  
  "name": "John Doe",
```

```
"email": "user@example.com",  
"password": "mypassword"  
}
```

Response:

```
{  
  "token": "jwt-token",  
  "user": {  
    "id": "123",  
    "name": "John Doe",  
    "email": "user@example.com"  
  }  
}
```

POST /api/auth/login – User login

Request:

```
{  
  "email": "user@example.com",  
  "password": "mypassword"  
}
```

Response:

```
{  
  "token": "jwt-token",  
  "user": {  
    "id": "123",
```

```
"name": "John Doe",  
"email": "user@example.com"  
}  
}
```

JWT tokens returned here are used to authorize requests to protected endpoints in SmartDocQ.

In short:

.env files created for staging and production.

Central config.js implemented for safe, centralized configuration.

API documentation updated for authentication and key endpoints.

API documentation for Authentication, AI Q&A, and File Management endpoints.

Setup instructions written for developers to run SmartDocQ in any environment.