**Project Documentation: Stile.it-Tile Management System**

**1. Project Overview**

The **Tile Management System** is a web-based application designed to streamline the organization, categorization, and display of various tiles used in interior design. The system allows Admins to manage tile records (CRUD operations), while Viewers can explore available tile options.

**2. Technology Stack**

* **Frontend:** React.js + Tailwind CSS
* **Backend:** .NET Core Web API
* **Database:** MySQL (Hosted on FreeSQLDatabase)
* **Authentication:** JWT (Role-based: Admin & Viewer)
* **Deployment:**
  + Backend → Render
  + Frontend → Vercel

**3. Roles and Functionalities**

* **Admin**
  + Login only (credentials hardcoded or manually added to DB)
  + Add / Edit / Delete Products, Categories, Applications
  + Block/Unblock entries
  + View activity logs
* **Viewer**
  + Can Register & Login
  + View only unblocked tiles, categories, applications

**4. Database Setup**

* Hosted on: [freesqldatabase.com](https://www.freesqldatabase.com/)
* Tables:
  + Users (with Role field as Admin or Client)
  + Products (linked with Category & Application)
  + Categories
  + Applications
  + ActivityLogs

**Note:** Admin is added manually in DB. Viewers can register via the frontend.

**5. Backend Setup (.NET Core)**

* API project created using Visual Studio / CLI
* Controllers for Auth, Products, Categories, Applications, Logs
* JWT Auth configured in Program.cs
* CORS configured for frontend deployment
* Dockerized using custom Dockerfile
* Environment variables configured on Render (Connection string, JWT secret)

**6. Frontend Setup (React + Tailwind)**

* Created using create-react-app
* Tailwind integrated via postcss
* Protected routing based on JWT roles
* Pages:
  + Login / Register
  + Admin Dashboard (Products, Categories, Applications)
  + Viewer Dashboard (Read-only)

Backend URLs updated manually in components (no .env used)

**7. Deployment Summary**

* **Backend**
  + Dockerized and deployed on [Render.com](https://render.com/)
  + Environment variables configured via Render Dashboard
* **Frontend**
  + Pushed to GitHub
  + Deployed to [Vercel.com](https://vercel.com/)
  + Build Command: npm run build
  + Output Directory: build

**8. Development & Deployment Issues – Solved**

**1. Backend Setup Errors-**

***Missing Tables in MySQL***

* **Issue:**  
  Error Table 'sql12786073.Users' doesn't exist when hitting login route.
* **Cause:**  
  Incorrect or incomplete SQL import.
* **Fix:**  
  Verified .sql file manually and re-imported corrected version via phpMyAdmin.

***SQL Syntax Error with Timestamps-***

* **Issue:**  
  MySQL threw error on datetime(6) type.
* **Fix:**  
  Changed datetime(6) to datetime to ensure compatibility across MySQL versions.

**2. Authentication & JWT Errors**

***Incompatible Package Versions***

* **Issue:**  
  Microsoft.AspNetCore.Authentication.JwtBearer 9.0.x is not compatible with net8.0
* **Fix:**  
  Downgraded all NuGet packages (JWT, OpenAPI) to 8.x versions in .csproj.

***JWT Issuer/Audience Mismatch***

* **Issue:**  
  401 Unauthorized errors on login despite correct credentials.
* **Fix:**  
  Matched values of Jwt:Issuer and Jwt:Audience in appsettings.json and frontend.

**3. Docker & Build Errors**

***Incorrect Dockerfile Paths***

* **Issue:**  
  lstat ... TileManagementSystem.csproj: no such file or directory
* **Fix:**  
  Updated Dockerfile to reflect exact folder structure. Used:
* COPY Backend/TileManagementSystem/TileManagementSystem.csproj ./TileManagementSystem/

***SDK Version Mismatch***

* **Issue:**  
  Render failed build: .NET SDK does not support targeting .NET 9.0
* **Fix:**  
  Changed project target framework to net8.0 in .csproj.

**4. Deployment Errors**

***MySQL Transient Connection Failure***

* **Issue:**  
  Unable to connect to any of the specified MySQL hosts
* **Cause:**  
  Free hosting (like FreeSQLDatabase) sometimes times out.
* **Fix:**  
  Added EnableRetryOnFailure() to UseMySql() in Program.cs.

***Missing Environment Variables on Render***

* **Issue:**  
  App crashes or returns 500 without error clarity.
* **Fix:**  
  Used double-underscore \_\_ in Render's Environment tab for nested config:
  + ConnectionStrings\_\_DefaultConnection
  + Jwt\_\_Key

**5. CORS & Frontend Issues**

***CORS Policy Blocked by Browser***

* **Issue:**  
  Access-Control-Allow-Origin header missing error from Vercel frontend.

**Fix:**  
Allowed all origins in backend:

builder.Services.AddCors(options =>

{

options.AddPolicy("AllowFrontend", policy =>

{

policy.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod();

});

});

app.UseCors("AllowFrontend");

***Hardcoded API URLs Causing Local/Prod Conflicts***

* **Issue:**  
  API calls worked locally but failed after deployment.
* **Fix:**  
  Replaced all localhost URLs with deployed Render API base URL.

**8. Conclusion**

* The system supports basic activity logging.
* All endpoints are protected with JWT.
* Viewer registration auto-redirects after success.
* No Admin registration route to prevent privilege escalation.

**9. Possible Future Enhancements**

* Image upload optimization
* Export data to Excel/PDF
* Add analytics for product popularity
* Pagination and search filters