**Cross Platform App Development Lab Experiment No. 6**

**Aim:** Integrating third party components into our application and XAML pages using shared resources.

**Objectives:**

1. Understand the process of integrating third-party components.

2. Explore the use of shared resources in XAML pages.

**Theory:**

- Integrating Third-Party Components:

- Involves adding pre-built components from external sources.

- Enhances functionality without building everything from scratch.

- Requires proper integration and configuration.

- Shared Resources in XAML:

- Resources shared across multiple XAML pages.

- Can include styles, templates, and other reusable elements.

- Promotes consistency in design and functionality.

**Requirements:**

- Access to the third-party component library.

- Existing XAML-based application.

**Tools:**

- Visual Studio or any XAML-compatible IDE.

- Necessary packages or tools for integrating third-party components.

**Implementation/ Code:-**

const path = require('path');

module.exports = ({ env }) => {

const client = env('DATABASE\_CLIENT', 'sqlite');

const connections = {

mysql: {

connection: {

connectionString: env('DATABASE\_URL'),

host: env('DATABASE\_HOST', 'localhost'),

port: env.int('DATABASE\_PORT', 3306),

database: env('DATABASE\_NAME', 'strapi'),

user: env('DATABASE\_USERNAME', 'strapi'),

password: env('DATABASE\_PASSWORD', 'strapi'),

ssl: env.bool('DATABASE\_SSL', false) && {

key: env('DATABASE\_SSL\_KEY', undefined),

cert: env('DATABASE\_SSL\_CERT', undefined),

ca: env('DATABASE\_SSL\_CA', undefined),

capath: env('DATABASE\_SSL\_CAPATH', undefined),

cipher: env('DATABASE\_SSL\_CIPHER', undefined),

rejectUnauthorized: env.bool(

'DATABASE\_SSL\_REJECT\_UNAUTHORIZED',

true

),

},

},

pool: { min: env.int('DATABASE\_POOL\_MIN', 2), max: env.int('DATABASE\_POOL\_MAX', 10) },

},

mysql2: {

connection: {

host: env('DATABASE\_HOST', 'localhost'),

port: env.int('DATABASE\_PORT', 3306),

database: env('DATABASE\_NAME', 'strapi'),

user: env('DATABASE\_USERNAME', 'strapi'),

password: env('DATABASE\_PASSWORD', 'strapi'),

ssl: env.bool('DATABASE\_SSL', false) && {

key: env('DATABASE\_SSL\_KEY', undefined),

cert: env('DATABASE\_SSL\_CERT', undefined),

ca: env('DATABASE\_SSL\_CA', undefined),

capath: env('DATABASE\_SSL\_CAPATH', undefined),

cipher: env('DATABASE\_SSL\_CIPHER', undefined),

rejectUnauthorized: env.bool(

'DATABASE\_SSL\_REJECT\_UNAUTHORIZED',

true

),

},

},

pool: { min: env.int('DATABASE\_POOL\_MIN', 2), max: env.int('DATABASE\_POOL\_MAX', 10) },

},

postgres: {

connection: {

connectionString: env('DATABASE\_URL'),

host: env('DATABASE\_HOST', 'localhost'),

port: env.int('DATABASE\_PORT', 5432),

database: env('DATABASE\_NAME', 'strapi'),

user: env('DATABASE\_USERNAME', 'strapi'),

password: env('DATABASE\_PASSWORD', 'strapi'),

ssl: env.bool('DATABASE\_SSL', false) && {

key: env('DATABASE\_SSL\_KEY', undefined),

cert: env('DATABASE\_SSL\_CERT', undefined),

ca: env('DATABASE\_SSL\_CA', undefined),

capath: env('DATABASE\_SSL\_CAPATH', undefined),

cipher: env('DATABASE\_SSL\_CIPHER', undefined),

rejectUnauthorized: env.bool(

'DATABASE\_SSL\_REJECT\_UNAUTHORIZED',

True ), },

schema: env('DATABASE\_SCHEMA', 'public'),},

pool: { min: env.int('DATABASE\_POOL\_MIN', 2), max: env.int('DATABASE\_POOL\_MAX', 10) },

},

sqlite: {

connection: {

filename: path.join(

\_\_dirname,

'..',

env('DATABASE\_FILENAME', '.tmp/data.db') ),},

useNullAsDefault: true,}};

return {

connection: {

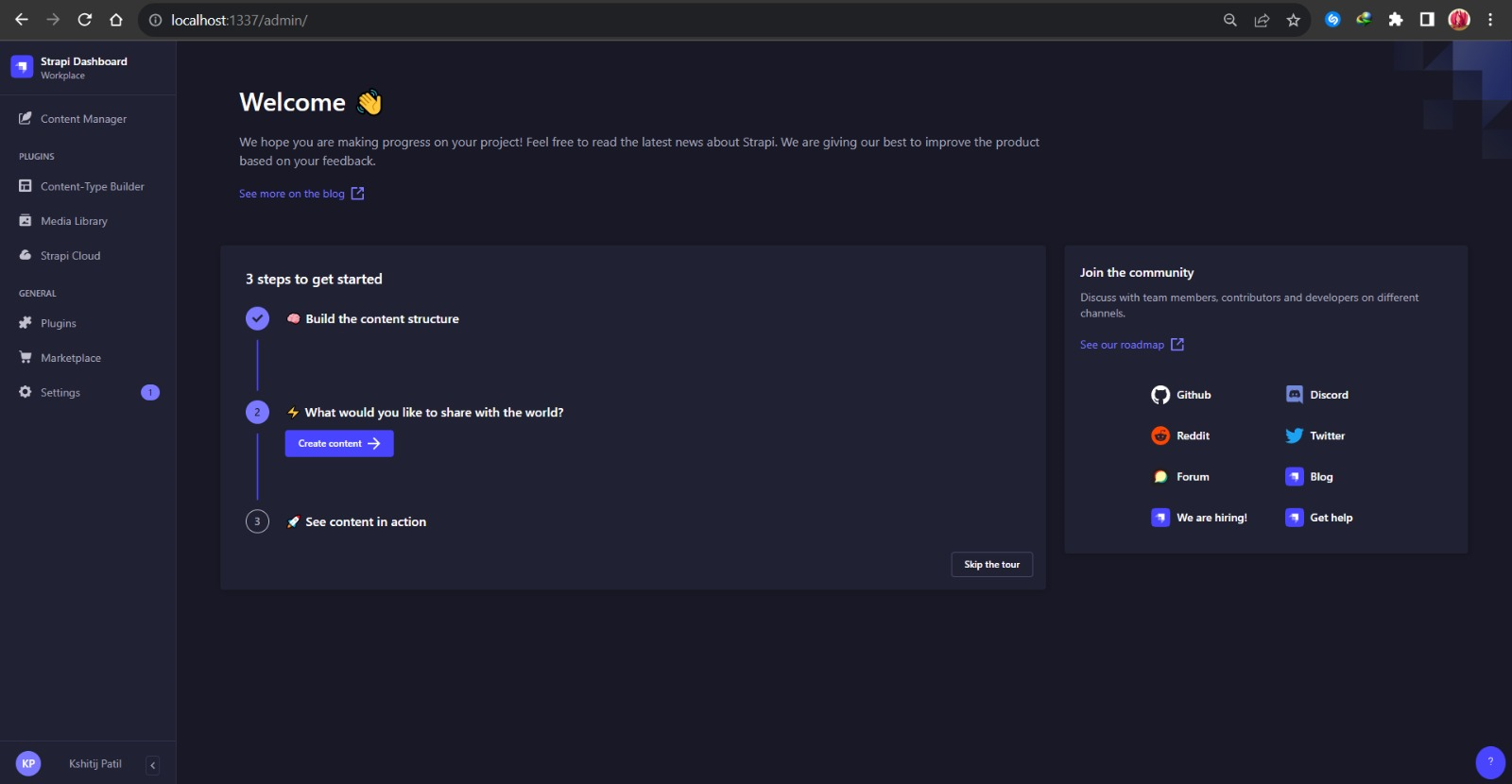
client,

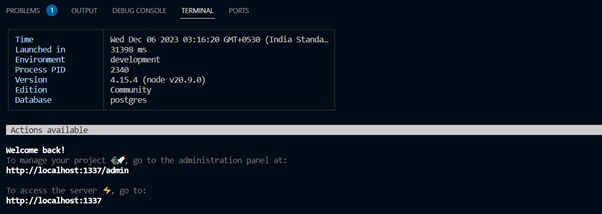
...connections[client],

acquireConnectionTimeout: env.int('DATABASE\_CONNECTION\_TIMEOUT', 60000),

},

};





**Conclusion:**

We successfully integrated third-party components and shared resources in XAML pages, we design of our application. This approach helps us save development time and promotes consistency in the user interface.

**References:**

1. **Microsoft XAML documentation:** [https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml](https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml)