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:

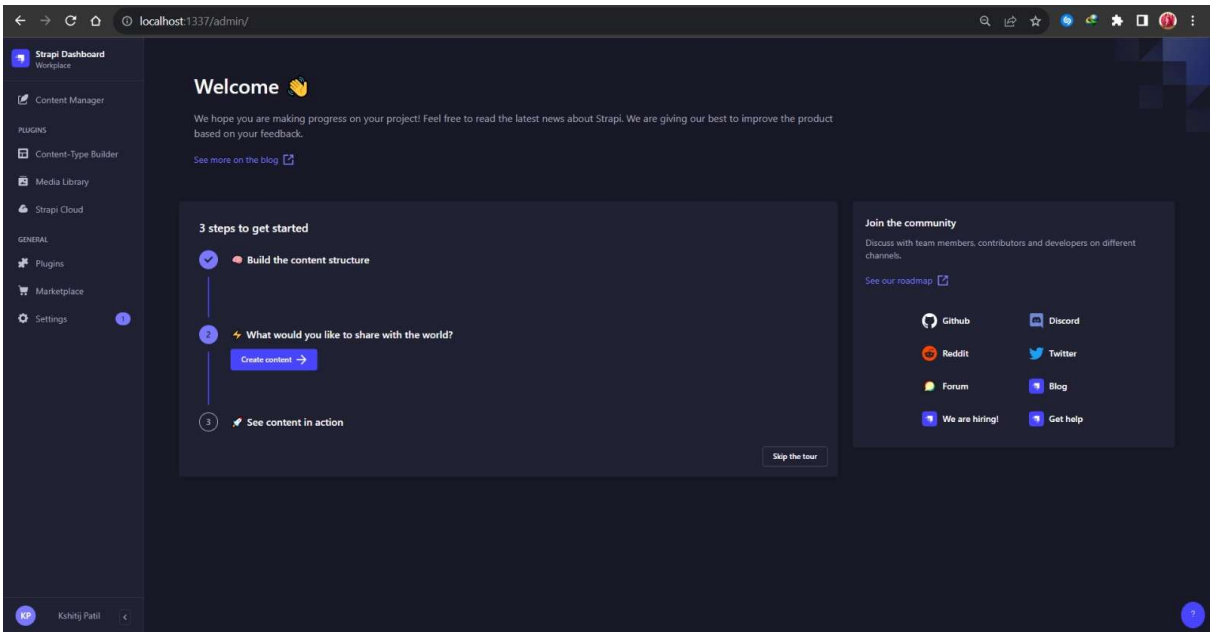
- Visual Studio or any XAML-compatible IDE.
- Necessary packages or tools for integrating third-party components.

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