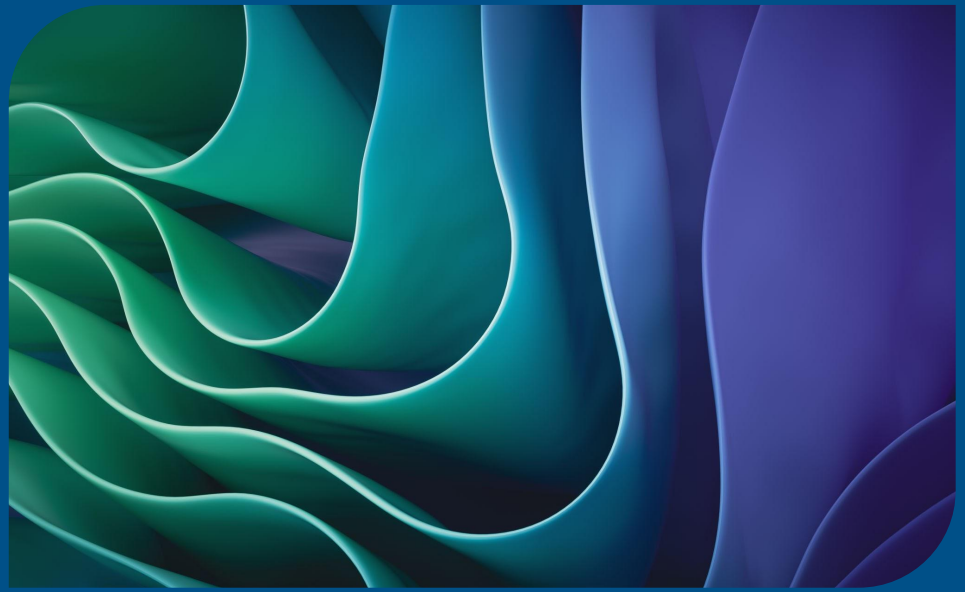
MM.DD.YY

Cross Platform Application Development Assignment SEZG585

Varun S
2023TM93663



Problem Statement

In today's digital age, many individuals seek a private and personalized space to express their creativity through writing, whether in the form of stories, poems, or other literary content. However, most existing platforms prioritize public sharing and social engagement, making it challenging for users who desire a private, distraction-free environment to write and organize their thoughts securely.

This application addresses the need for a **secure, user-centric platform** where individuals can write, save, and manage their creative writings—such as stories and poems—**privately**. Each user's content is stored and listed exclusively for them, ensuring complete ownership and confidentiality. By eliminating the pressure of public validation, the app fosters a comfortable and inspiring space for authentic self-expression and personal growth.

Objectives

The objective of this application is to provide a **secure and user-friendly platform** that enables individuals to **write, store, and manage their creative content**—such as stories and poems—in a **private and personalized environment**. The app aims to:

1. **Encourage creative expression** by offering a distraction-free space for writing.
2. **Ensure user privacy** by storing content such that it is only accessible to the respective user.
3. **Provide intuitive content management** features for organizing and viewing written works.
4. **Enhance user experience** through a clean, responsive interface optimized for writing.
5. **Support personal growth and reflection** by allowing users to privately track their creative journey over time.

Source Code and Demo Link

Demo Link : https://youtu.be/_WBqFJA4-C0

Source code: https://github.com/VarunRocky007/cp-assignment

Back4App architecture

Back4App is a **Backend-as-a-Service (BaaS)** platform that provides developers with an easy way to build, host, and scale apps without having to manage the backend infrastructure themselves. It is based on **Parse Server**, an open-source backend framework originally developed by Facebook. Here's a high-level overview of the typical **Back4App backend architecture**:

Core Components of Back4App Backend Architecture

1. Parse Server (Application Layer)

- This is the heart of the Back4App backend.
- Written in Node.js and runs on Express.js.
- Provides REST and GraphQL APIs.
- Manages business logic via:
    - Cloud Functions
    - Triggers (before/after save, delete, etc.)
    - Jobs (background tasks)

Back4App architecture

2. Database Layer

- Typically uses **MongoDB** (NoSQL) to store application data.
- Back4App handles scaling, backups, and replication.
- Supports relational data modeling via pointers and relations.

3. Authentication & Security

- Built-in user authentication (email/password, social logins).
- Supports roles and ACLs (Access Control Lists) for data access control.
- App-level security settings like class-level permissions and app keys.

Back4App architecture

4. Real-time Engine

- Based on **WebSockets**.
- Enables real-time data updates through **Live Queries**.

5. File Storage

- Files are stored using a third-party provider (e.g., AWS S3).
- Files are accessed via URLs and managed by Parse File API.

6. Hosting & Deployment

- Apps run in a managed cloud environment.
- Developers can deploy custom Parse Server versions via containers (Docker) or node environments.
- Provides CLI and GitHub integration for CI/CD workflows.

7. Dashboard (Admin Panel)

- Web-based GUI for managing your backend (data, users, classes, jobs, etc.).
- Based on the open-source Parse Dashboard.

Back4App integration with Flutter

Flutter connects to **Back4App** (which runs Parse Server) through an official Flutter SDK (`parse_server_sdk_flutter`). This SDK allows your Flutter app to perform backend operations like authentication, data storage, queries, and more.

Key Features of Flutter + Back4App Integration

1. Initialization

Before any backend operation, your Flutter app must initialize the Parse SDK with credentials from your Back4App dashboard—namely, your Application ID, Client Key, and Server URL.

Back4App integration with Flutter

2. User Authentication

Back4App supports user authentication out of the box. Your Flutter app can handle:

- **User sign-up**: Creating a new user with a username, email, and password.
- **Login**: Authenticating existing users.
- **Session management**: Maintaining login sessions via tokens.
- **Logout**: Ending user sessions securely.
- **Password reset**: Sending password recovery emails.
- **Email verification**: Enforcing verified emails before login (if enabled).

Back4App integration with Flutter

3. Data Management

Flutter can create, read, update, and delete backend objects (CRUD operations). Back4App uses MongoDB on the backend, and data is stored in classes (similar to tables).

Your app interacts with this data by:

- Defining object classes
- Storing/retrieving field values
- Running queries and filters
- Handling pagination and sorting

Back4App integration with Flutter

4. Security

Back4App includes robust security controls like:

- **Access Control Lists (ACLs)**: Define which users or roles can access specific records.
- **Class-Level Permissions (CLPs)**: Manage what operations are allowed on entire classes.
- **Roles**: Group users and assign shared permissions.

These settings help protect user data and restrict access where necessary.

Thank You