

# Project Overview

## Task Manager App

- ★ User Authentication (Register/Login with session management)
- ★ Complete CRUD Operations (Create, Read, Update, Delete tasks)
- ★ Cloud Backend Integration (Back4App Parse Server)
- ★ Neumorphic UI Design (3D depth effects)

## Tech Stack

- ★ Frontend: Flutter 3.9.2, flutter\_neumorphic\_plus, Google Fonts
- ★ Backend: Back4App (BaaS), Parse Server SDK
- ★ Design: Neumorphic/Skeuomorphic UI patterns

# Architecture & Implementation

## System Architecture

Flutter App (Neumorphic UI)

- + Parse SDK

Back4App Cloud (Parse Server)

- Authentication Service
- Task Database
- ACL Security

## Key Implementation

- ★ ParseObject subclass registration for Task model
- ★ ACL permissions for user-specific data access
- ★ Session validation and error handling

## Project Structure

```
lib/
    └── main.dart          # App initialization, Parse setup
    └── models/task.dart   # Task ParseObject model
        └── screens/
            ├── login_screen.dart      # Email/password authentication
            ├── register_screen.dart   # User registration
            ├── task_list_screen.dart  # Read, Delete operations
            └── add_edit_task_screen.dart # Create, Update operations
...
```

# CRUD Operations & Features

## Complete Task Management System

| Operation | Implementation | Features |

|-----|-----|-----|

| \*\*Create\*\* | Add new tasks | · Required title (max 100 chars)<br>· Optional description (max 500 chars)<br>· Character counter<br>

ACL security setup |

| \*\*Read\*\* | View task list | · User-specific query filtering<br>· Sorted by creation date<br>· Pull-to-refresh sync<br>· Session validation |

| \*\*Update\*\* | Edit tasks | · Modify title/description<br>· Form validation<br>· Error handling with Snackbar |

| \*\*Delete\*\* | Remove tasks | · Confirmation dialog<br>· Success/error feedback<br>· Immediate UI update |

## Authentication Flow

- \*\*Register:\*\* Email + password validation, confirmation matching
- \*\*Login:\*\* Session management, auto-redirect to task list
- \*\*Logout:\*\* Session cleanup, confirmation dialog
- \*\*Security:\*\* ParseACL for user-specific data access

# Neumorphic UI Design

## Design Philosophy

Neumorphic Design\*\* = Soft UI with realistic 3D depth and shadows

## Visual Implementation

### Depth System:\*\*

- Raised elements: depth 4-8 (buttons, cards)
- Inset elements: depth -6 to -3 (input fields)
- Completed tasks: depth -4 (pressed effect)

### Components:\*\*

- 3D circular icons with concave depth
- Inset text fields (carved surface effect)
- Raised buttons with shadow layers
- Neumorphic cards for tasks

### Color Scheme:\*\*

- Base: `#E0E5EC` (soft gray background)
- Primary: Deep Purple `#673AB7` (login, main actions)
- Success: Green `#66BB6A` (register, confirmations)
- Error: Red `#EF5350` (errors, delete)

# Application Screens

## 1. Login Screen

- 3D circular icon with concave depth
- Inset email/password fields
- Raised purple login button
- Password visibility toggle
- Navigation to register

## 3. Task List Screen

- App bar with icon, "Tasks" title, refresh & logout buttons
- Neumorphic task cards (depth 8 for active, -4 for completed)
- Edit (blue) and Delete (red) buttons per task
- Empty state with 3D icon and message
- Purple floating action button (+)

## 2. Register Screen

- Three validation fields (email, password, confirm)
- Green raised button
- Success dialog with animation
- Automatic redirect after registration

## 4. Add/Edit Task Screen

- Info card with instructions
- Inset title field (max 100 chars)
- Inset description field (max 500 chars, 6 lines)
- Real-time character counters
- Raised save/update button

# Technical Implementation

## Back4App Integration

```
```dart
// Parse initialization with Task registration
await Parse().initialize(
  keyApplicationId, keyParseServerUrl,
  clientKey: keyClientKey,
  registeredSubClassMap: {'Task': () => Task()},
);

// ACL Security for user-specific data
final acl = ParseACL(owner: currentUser);
acl.setPublicReadAccess(allowed: false);
task.setACL(acl);

// Query user's tasks only
final query = QueryBuilder<Task>(Task())
..whereEqualTo('createdBy', currentUser)
..orderByDescending('createdAt');
```

## Data Model

```
```dart
Task {
  title: String (required, max 100)
  description: String (optional, max 500)
  isCompleted: Boolean
  createdBy: Pointer<User>
  ACL: User-specific permissions
}
```

```

# Key Learnings & Results

## Technical Skills Gained

- **BaaS Integration:** Parse Server SDK, cloud database, authentication
- **Flutter Development:** StatelessWidget, form validation, async/await, navigation
- **UI/UX Design:** Neumorphic principles, depth manipulation, color theory
- **Security:** ACL implementation, session management, input validation

## Challenges Solved

- Package compatibility (`flutter_neumorphic_plus` migration)
- ParseObject subclass registration
- Layout overflow fixes (`Flexible`, `Wrap`, `TextOverflow`)
- Session validation & error feedback

## Project Highlights

Fully functional CRUD with error handling

-  Secure user authentication & data isolation
-  Beautiful neumorphic UI with consistent design
-  Responsive layouts without overflow errors
-  Production-ready with proper validation

# resources and links

Github:[https://github.com/kanakapalli/bits\\_assignment\\_flutter\\_task\\_manager\\_app\\_with\\_back4app](https://github.com/kanakapalli/bits_assignment_flutter_task_manager_app_with_back4app)

video demo :

[https://github.com/kanakapalli/bits\\_assignment\\_flutter\\_task\\_manager\\_app\\_with\\_back4app/blob/master/Demo.mp4](https://github.com/kanakapalli/bits_assignment_flutter_task_manager_app_with_back4app/blob/master/Demo.mp4)

email: 2024tm93318@wilp.bits-pilani.ac.in

