**SmartLife+ — Hibrid Mobil Tətbiq**

## **Layihənin Modul Strukturu**

|  |  |
| --- | --- |
| **Modul** | **Texnologiya** |
| 📝 Qeydlər (Notes) | AsyncStorage |
| ✅ Tapşırıqlar (Tasks) | SQLite |
| ☁️ Hava (Weather) | REST API (OpenWeatherMap və ya simulyasiya) |
| 📰 Xəbərlər (News) | REST API (JSONPlaceholder / newsapi) |
| 👤 Profil | AsyncStorage (avatar, ad, dil) |
| ⚙️ Ayarlar (Tema) | useContext + AsyncStorage |
| 🔍 Axtarış və Filter | JavaScript + FlatList |
| 🔄 İnternet yoxdursa Offline oxuma | AsyncStorage backup |

## **🎯 Ümumi Məqsəd:**

İstifadəçiyə öz gündəlik həyatını idarə etmək üçün:

* **tapşırıqlar yazmaq**
* **qeydlər aparmaq**
* **hava məlumatını izləmək**
* **xəbərləri oxumaq**
* **profilini və ayarlarını saxlamaq**

imkanını verən **mobil tətbiq**dir.

## **🧱 Əsas Modul və Ekranlar**

### **1️⃣ HomeScreen (Baş səhifə)**

* 4 əsas modulun shortcut card-ları:  
  + Tapşırıqlar
  + Qeydlər
  + Hava
  + Xəbərlər
* Ekranın yuxarısında Salam, [ad] yazısı (profil məlumatından gəlir)
* Drawer menyunu açmaq üçün hamburger menyu

### **2️⃣ Tasks Module (SQLite)**

|  |  |
| --- | --- |
| **Ekran** | **Funksiyalar** |
| TasksListScreen | Bütün tapşırıqları siyahılayır (FlatList), tamamlandı statusu göstərilir |
| AddTaskScreen | Yeni tapşırıq əlavə etmək (başlıq, deadline) |
| TaskDetailScreen | Mövcud tapşırığı düzəltmək və silmək |
| Filter Toggle | Tamamlanmış və tamamlanmamış tapşırıqları ayırmaq |
| SQLite | tasks cədvəli yaradılır və saxlanılır |

### **3️⃣ Notes Module (AsyncStorage)**

|  |  |
| --- | --- |
| **Ekran** | **Funksiyalar** |
| NotesListScreen | Qeydləri siyahılayır (başlıq görünür) |
| AddNoteScreen | Qeyd əlavə et (başlıq və məzmun) |
| NoteDetailModal | Qeydin məzmununa baxmaq və silmək |
| **AsyncStorage** | **notes açarı altında saxlanılır**  **Mueyyen saatda alert kimi qeyd xatirlatmasini telefona bildirish gondermelidi** |

### **4️⃣ Weather Module (REST API + Cache)**

|  |
| --- |
| **Funksiyalar** |
| Şəhər adı daxil edildikdən sonra hava məlumatı çəkilir (OpenWeatherMap və ya fake API ilə) |
| Məlumat göstərilir: temperatur, hava statusu, ikon |
| Axırıncı nəticə AsyncStorage-də saxlanılır |
| İnternet yoxdursa, son məlumat göstərilir |
| fetch() və ya axios ilə HTTP GET işlənir |

### **5️⃣ News Module (REST API)**

|  |
| --- |
| **Funksiyalar** |
| Xəbərləri list şəklində göstərir (https://jsonplaceholder.typicode.com/posts) |
| Xəbər üzərinə kliklə NewsDetailScreenə keçid |
| Lazy loading (istəyə görə FlatList onEndReached) |
| AsyncStorage backup (optional) |
| useEffect + fetch ilə API çağırışı |

### **6️⃣ Profile Module (AsyncStorage + useContext)**

|  |
| --- |
| **Funksiyalar** |
| Ad, soyad, bio və avatar emoji daxil edilir |
| AsyncStorage-də saxlanılır, hər açılışda yüklənir |
| useContext ilə digər modullara istifadəçi adı paylaşılır (məs: HomeScreen-də Salam, Elvin) |

### **7️⃣ Settings Module**

|  |
| --- |
| **Funksiyalar** |
| Tema seçimi: Light və Dark |
| Saxlanma: AsyncStorage ilə son tema seçimi yadda saxlanır |
| useContext ilə app-ə yayılır |
| Dil seçimi |
| Logout düyməsi: AsyncStorage.clear() edir və app “reset” olur |

### **🔐 Drawer Menu**

* Soldan açılır
* Aşağıdakılar olacaq:  
  + Profil
  + Ayarlar
  + Logout

## **🧪 İstifadəçi Axını (User Flow)**

### **✅ 1. İlk açılış:**

* AsyncStorage yoxlanır → ad varsa, salamlanır
* Əks halda Profil ekranına yönləndirilə bilər

### **✅ 2. İstifadəçi tapşırıq əlavə edir:**

* Tasks ekranında “+” düyməsi ilə AddTaskScreen-ə keçid
* SQLite-də yadda saxlanır
* Siyahıya qayıdanda yeni task görünür

### **✅ 3. İstifadəçi qeyd əlavə edir:**

* Notes modulunda başlıq + məzmun daxil edilir
* AsyncStorage-ə yazılır

### **✅ 4. İstifadəçi hava ekranına keçid edir:**

* Şəhər daxil edir → API məlumatını çəkir
* Offline olsa, son məlumat AsyncStorage-dən göstərilir

### **✅ 5. İstifadəçi xəbərlərə baxır:**

* API-dən dinamik data gəlir
* navigation.navigate('NewsDetail') ilə detala keçid

### **✅ 6. Ayarlar ekranında:**

* Tema dəyişir → Context vasitəsilə bütün tətbiqə yayılır
* AsyncStorage ilə saxlanılır