

A Major Project Synopsis on

# **PrimeNest**

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

**MASTER OF COMPUTER APPLICATIONS**

2023-2025

by

Mihir

23FS20MCA00008



**MANIPAL UNIVERSITY  
JAIPUR**

Under the guidance of

Dr. Pragya Vaishnav

**Department of Computer Applications**

**School of AIML, IoT&IS, CCE, DS and Computer Applications**

**Faculty of Science, Technology and Architecture**

**Manipal University Jaipur**

**Jaipur, Rajasthan**

**2025**

## **I. Introduction**

PrimeNest is a modern, full-stack real estate mobile application built using React Native and Expo SDK 52. It provides a seamless experience for users looking to buy, rent, or explore properties while leveraging the latest mobile technologies for fast, responsive, and visually appealing interfaces.

The app includes essential real estate features such as secure authentication with Google, advanced search and filtering options, property details with images, and a personalized user profile.

Why should you use PrimeNest?

### **1. For Property Seekers**

- a. Browse a wide range of properties from various categories
- b. Apply filters for location, budget, and property type
- c. View detailed property pages with images, descriptions, and amenities
- d. Save favorite properties for easy access

### **2. For Real Estate Agents & Property Owners**

- a. List properties with images, details, and pricing
- b. Manage property listings easily through an intuitive interface
- c. Attract potential buyers and renters with a sleek and user-friendly platform

## **II. Motivation**

The real estate industry often struggles with outdated applications, clunky interfaces, and inefficient search mechanisms. PrimeNest addresses these issues by offering:

### **1. For Users**

- a. A mobile-friendly real estate platform that simplifies property discovery
- b. A seamless and visually engaging browsing experience with Tailwind CSS for styling
- c. Easy authentication and profile managementSave favorite properties for easy access

### **2. For Agents & Property Owners**

- a. A structured system to manage property listings efficiently
- b. A scalable backend powered by Appwrite for handling authentication, data storage, and API management
- c. A streamlined process for showcasing properties with high-quality images and details

## **III. Problem Statement**

### **1. Challenges in Existing Real Estate Applications**

- a. Outdated user interfaces that are difficult to navigate
- b. Poor search and filtering options leading to inefficient property discovery
- c. Lack of seamless authentication and data security
- d. No easy way for property owners to list and manage properties

## 2. How PrimeNest Solves These Problems

- a. Uses React Native and Expo to create a smooth and interactive UI
- b. Implements dynamic routing and state management for seamless navigation
- c. Uses Google authentication for secure and hassle-free login
- d. Optimizes data fetching with TanStack Query to improve performance
- e. Provides a clean, type-safe codebase using TypeScript

## IV. Methodology/ Planning of work:

1. UI Development with React Native, the application will have five primary screens:
  - a. Home Page – Displays featured and recommended properties
  - b. Explore Page – Allows users to browse different property types
  - c. Property Details Page – Shows in-depth details of selected properties
  - d. Profile Page – Lets users manage settings and saved properties
  - e. Authentication Page – Handles user login and signup
2. Backend & Data Management with Appwrite
  - a. Authentication: Secure Google login for seamless user access
  - b. Database Architecture: Scalable backend for storing property listings and user data
  - c. Real-time Data Fetching: Uses TanStack Query to manage API requests efficiently
3. Navigation & Routing
  - a. Implements Expo Router for file-based routing
  - b. Uses tab-based navigation for an intuitive user experience
  - c. Supports deep linking for direct access to specific properties
4. Styling & User Experience
  - a. Uses Tailwind CSS and NativeWind for fast and responsive UI
  - b. Implements custom themes, fonts, and color schemes

## V. Requirements for proposed work:

1. Software Requirement:
  - a. Oper Operating System: Windows / macOS / Linux
  - b. Frontend: React Native with Expo SDK 52
  - c. Backend: Appwrite (Backend-as-a-Service)
  - d. Database: Appwrite's integrated database
  - e. Styling: Tailwind CSS with NativeWind
  - f. Routing & Navigation: Expo Router
2. Hardware Requirement:
  - a. Processor: Minimum Intel i3 or equivalent

- b. RAM: 4GB or higher
- c. Storage: 10GB free space for development

## **VI. Bibliography/References**

- Official React Native documentation – <https://reactnative.dev>
- Expo SDK 52 – <https://expo.dev>
- Appwrite Documentation – <https://appwrite.io/docs>
- Tailwind CSS for React Native – <https://tailwindcss.com>
- TanStack Query (React Query) – <https://tanstack.com/query>
- React Native Reanimated (for Animations & UI Interactions)  
<https://docs.swmansion.com/react-native-reanimated>
- React Native Paper (UI Components Library) – <https://reactnativepaper.com>
- React Native Firebase (Alternative Backend & Authentication) – <https://rnfirebase.io>