# 📄 Project Brief / Prompt: InGermany App (обновлено)

## 🔹 Общая информация

**Автор проекта:** Umed Sabzaev (Германия, Тюрингия)  
**Название приложения:** **InGermany – Work, Life and Study**  
**Категория:** Справочник / Путешествия / Жизнь за границей  
**Цель:** помочь мигрантам, студентам и семьям, планирующим переезд в Германию, разобраться в жизни, работе, бюрократии и финансах.

## 🔹 Текущий статус проекта (13.09.2025)

* Создан проект **InGermany** в **Xcode** (Swift + SwiftUI, iOS 17+).
* Настроена структура папок:
  + **Core/** → основной запуск приложения (InGermanyApp.swift, ContentView.swift).
  + **Models/** → модели данных (Article.swift, Category.swift).
  + **Views/** → экраны (HomeView.swift, CategoriesView.swift, ArticlesByCategoryView.swift, SearchView.swift, ArticleView.swift).
  + **Resources/** → данные в формате JSON (articles.json, categories.json).
  + **Services/** → загрузка данных (DataService.swift).
* Реализован прототип приложения с поддержкой локализации (русский / английский / таджикский).
* JSON-файлы подключены через **Copy Bundle Resources**, корректно загружаются при запуске.
* Сделаны базовые экраны: Home, Categories, ArticlesByCategory, Search, ArticleView.
* Поддержка **многоязычных статей и категорий** через вложенные объекты (ArticleText, CategoryName) и методы localizedTitle, localizedContent, localizedCategory, localizedName.
* Работает фильтрация статей по языку и категории.
* Поиск статей по заголовку, содержанию и тегам.
* Реализованы теги:
  + Отображаются в ArticleView и SearchView.
  + Подсветка совпадающих тегов.
  + Кликабельные теги → открывается поиск с фильтрацией по тегу.

## 🔹 Структура файлов и задачи

### Core/

* **InGermanyApp.swift** → Точка входа в приложение, настройка ModelContainer, запуск ContentView.
* **ContentView.swift** → Главное окно с TabView (Home, Categories, Search, Settings).

### Models/

* **Article.swift** → Модель статьи, поддержка многоязычности (RU/EN/TJ), методы localizedTitle, localizedContent, localizedCategory.
* **Category.swift** → Модель категории (иконка + локализованные названия), метод localizedName.

### Views/

* **HomeView.swift** → список всех статей по выбранному языку.
* **CategoriesView.swift** → список категорий (Работа, Учёба, Финансы и т.д.) с иконками и локализацией.
* **ArticlesByCategoryView.swift** → статьи в выбранной категории, фильтрация по языку.
* **SearchView.swift** → поиск по заголовкам, содержимому и тегам, подсветка и кликабельность тегов.
* **ArticleView.swift** → экран с полной статьёй (заголовок, текст, категория, теги, переход по тегам).
* **SettingsView.swift** (планируется) → выбор языка приложения (RU/EN/TJ).

### Services/

* **DataService.swift** → загрузка JSON-файлов (articles.json, categories.json), парсинг в модели.

### Resources/

* **articles.json** → база статей (каждая статья хранится в 3 языках + теги).
* **categories.json** → список категорий с локализацией (RU/EN/TJ) и иконками.

## 🔹 Следующие шаги

* Добавить **оценку статей** (лайк / звёздочка).
* Добавить **возможность поделиться статьёй** (через стандартный iOS share sheet).
* Сделать **SettingsView** с переключением языка.
* Поддержка **регистрации и входа** (Email, Google, Apple ID).
* Личный кабинет: закладки, сохранённые статьи.
* Улучшение дизайна в стиле Booking / App Store.

## 🔹 Требования к коду

* Swift + SwiftUI, iOS 17+.
* Модульная структура.
* Каждый экран и функция подробно прокомментированы на русском языке.
* Имена файлов, классов и функций — в стиле международных iOS-разработчиков (CamelCase).
* Чистый код, готовый к демонстрации работодателю.