**FRETZApp (Facilitating Real-time Engaging Tap Zoned Application): User-Interactive AI Mobile App for Strand Selection and Career Information Advocacy**

***FRETZIE P. CAMBIADO***

**Subject: MOBILE COMPUTING**

**platform used: FLUTTER/ ANDROID STUDIO**

**date of submission: JULY 21, 2025**

**MASTER IN INFORMATION TECHNOLOGY**

## CHAPTER I

### **Introduction**

**Background of the Study**

The opportunity to be informed and be guided for career decisions have become the core elements of education across the global platform. Sahito (2025) found out that career education has to be inculcated within the academic discourse through immediate, interactive and engaging system which will align learners to the demands of actual world enterprise. This will extend significance to learners, educators and policy makers as to lead learners to a successful undertaking in the future. Globally, the incorporation of AI technology in career guidance posits improvement among students’ decision-making skills together with maintaining privacy in their challenges through a specific but timely approach to guidance in meeting their discourse needs (Saparovna, 2024). Westman (2021) found out that potential value and functions for artificial intelligence concerning the help to choose a career and align learners with appropriate information can be very vital in the life-long learning sustainability. Nevertheless, even with high rising technology, its core is only to admit the learner-entrants in school, however, their strands, or alignment to their individual prospected career attached to their skills are not apparent. Though Ai can provide technology engaged context for information, there is still but a space for Ai-generated mobile app, locally or inclusively developed to interactively provide chatbots, read-aloud artificial intelligence to facilitate career advocacy and institutional information. Furthermore, many remedies to access engagement but there is but limited research undertaking on senior high school students and teacher’s collaborative potential to be kept abreast of the culturally important and interactive features of mobile app that can be utilized for career selection and career advocacy which can reached out to individuals without even going online most of the time. In the researcher’s school information dissemination, information guidance and career guidance advocacy are highly needed because it is observed that learners are showing the need for interactive, Ai-driven career recommendations such as chatbots, read-aloud electronic landscapes for tailoring academic and social context. More so, the traditional undertaking for learners’ query and learning preferences tend to be very data limited and interactivity enhanced. According to Vieriu (2025) The significance of framework for AI implementation, backed by ethical standards, leverage the advantages while reducing risks. To conclude, although AI has great potential to improve learning efficiency and academic achievement, its effective implementation necessitates tackling issues concerning accuracy, cognitive disengagement, and ethical considerations. A well-rounded strategy is crucial to guarantee fair, efficient, and accountable learning opportunities in AI-driven educational settings.

Thus, using FRETZApp it aims to fill these gaps by delivering an interactive, accessible, real-time engagements for learners and educators with AI interactions to serve their queries which will not only keep abreast the demands of global and local market but as well instill culturally reach technology initiative, more specific technology that will serve an academic platform’s long time challenge for fast, informative, interactive and collaborative AI technology both online and offline opportunities can be served.

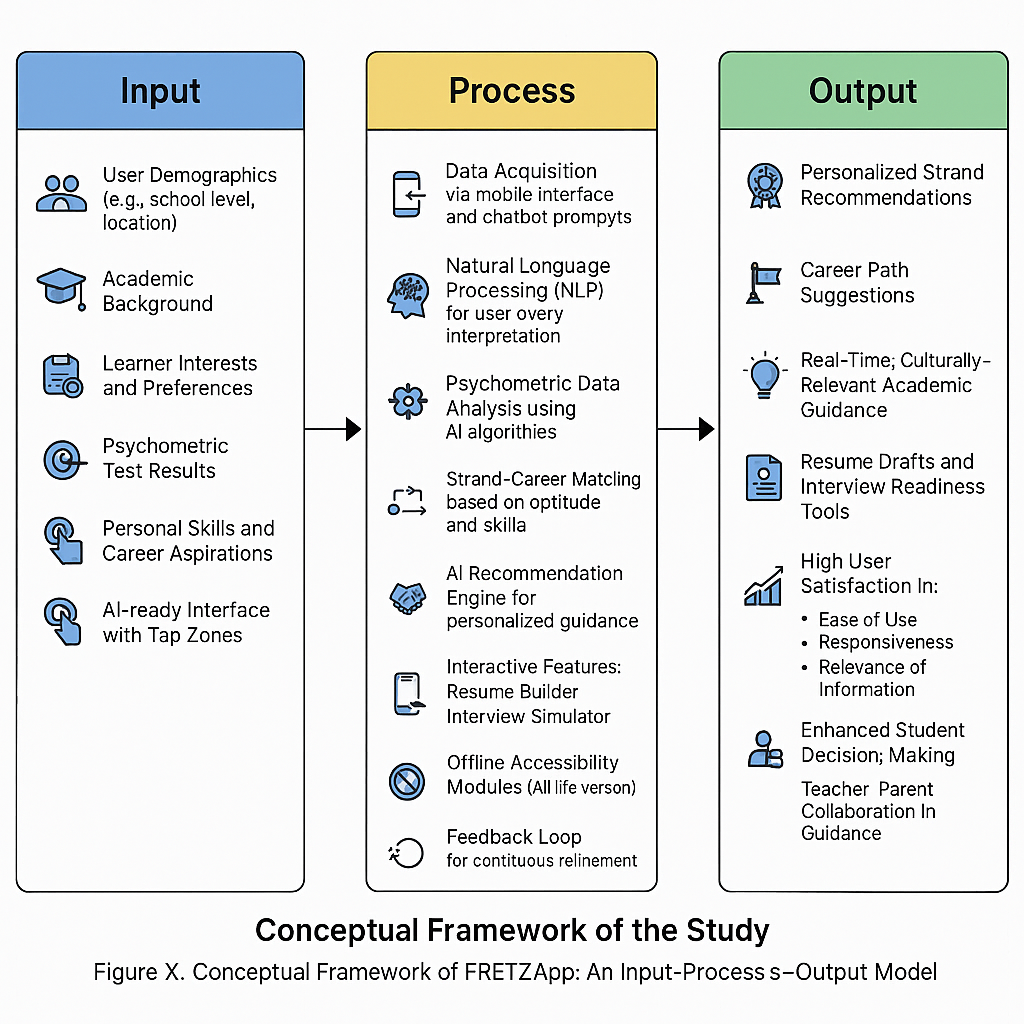
s

### **Theoretical Framework of the Study**

**Figure 1. Theoretical Framework**

This study has myriad of theoretical underpinnings which theoretically framed the FRETZApp as user-interactive AI on the preferences of career advocacy and information source on academic undertaking. Generally, The Constructivist Learning Theory (Piaget, 1970; Vygotsky, 1978) asserts that learners actively construct their own understanding through interaction with their environment. By using a real-time data source, AI communicative chatbots, students gain immediate feedback, career advise and information sources that will help them decide while maintaining their privacy in academic undertakings, encouraging them to take responsibility for their learning preferences. This theory supports the idea that students benefit from being actively involved in managing their academic journey.

The Human-Computer Interaction theory of Dix et. al (2004) which describe how query-based interfaces influencing the AI dynamics of human-computer interaction for career guidance application which as well emphasized Schuler and Namioka (1993) on the apps ability to provide the needed sources for learning outcomes. Likewise, theory on Intelligent Learning Platforms and Adaptive System as Brusilovsky (2001) emphasized on how educational data and guidance can provide learners the profile and choice which can lead to increased motivation and cognition. More so, can contribute to significant career decision skills. Furthermore, Social Cognitive Career Theory of Brown and Hackett 1994) sufficed the AI ability to personalize advice, privacy, confidence and skill development when learners are undergoing AI generated interview using the FretzApp. Finally, the use of Data Quality and AI Decision-Making as a framework that supports the neutral logistics and accurate AI recommendations as described by Floridi et al (2018).

**Figure 2. Conceptual Framework**

The figure above illustrated the determination of Inputs, Processes, and Outputs for the proposed FRETZApp. This is integral to trace the system is done in vivid understanding of the procedure. FRETZApp functions through a clear input-process-output system to provide personalized strand selection and career information advocacy.

Enclosed undertaking in the input stage, the app gathers important data form users, including their school background, interests, skills, and responses to psychometric assessments via an interactive mobile interface. Users participate with the app by tapping assigned zones in responding AI-generated reminders, which secure integral private career-related data. The users’ responses will be the the basis for the AI algorithms to analyze individual preferences and aptitudes.

Within the process phase, FRETZApp’s AI engine integrates digital learning in natural language processing to infer the input data, matching user profiles with appropriate academic strands and career pathways. It involves the real-time work market inclinations and skill requisites to provide appropriate recommendations. The system also offers features such as resume building and interview preparation to enhance user readiness. In the output phase, the app delivers personalized strand suggestions, detailed career insights, and actionable guidance through an engaging and user-friendly interface. This real-time, adaptive feedback undertakings contribute a specific decision-making and career planning, consistent with findings on AI-enhanced career guidance systems that improve accessibility and relevance for students (Westman et al., 2021; Kaldybaeva, 2025; Kulugh et al., 2025). Moreover, this system will highlight all access spectrum for efficient, proficient and culturally inclusive digital undertaking for all especially in Isulan National High School.

**General Objective**

Generally, this study generally aims to design, develop, and evaluate FRETZApp (Facilitating Real-time Engaging Tap Zoned Application), a user-interactive AI mobile application for strand selection and career information advocacy, to enhance students’ decision-making processes regarding academic strand choices and career planning.

**Specifically, this study seeks to answer:**

1. What is the level of user acceptance and satisfaction in using FRETZApp across the following:
   1. Content quality;
   2. Instructional design;
   3. Technical performance; and
   4. Over-all user experience?
2. What is the level of student decision making skills as regards to academic strand selection and career planning based on their pre-test and post-test survey scores?
3. Is there a significant difference between the student decision making skills regarding academic strand selection and career planning as measured by their pre and post survey scores?
4. Is there a significant relationship between system quality, information quality, mobile self-efficacy and actual user engagement?
5. Is there a significant difference in the user acceptability of FRETZApp across the three parameters of evaluation, security and privacy, accuracy, and transparency among students, teachers, and parents?

**Hypothesis**

In this study, the following hypotheses will be tested at 0.05 level of significance:

H01: There is no significant difference between the student decision making skills regarding academic strand selection and career planning as measured by their pre and post survey scores.

H02: There is no significant relationship between system quality, information quality, mobile self-efficacy and actual user engagement.

H03: There is no significant difference in the user acceptability of FRETZApp across the three parameters of evaluation, security and privacy, accuracy, and transparency among students, teachers, and parents.

**Significance of the Study**

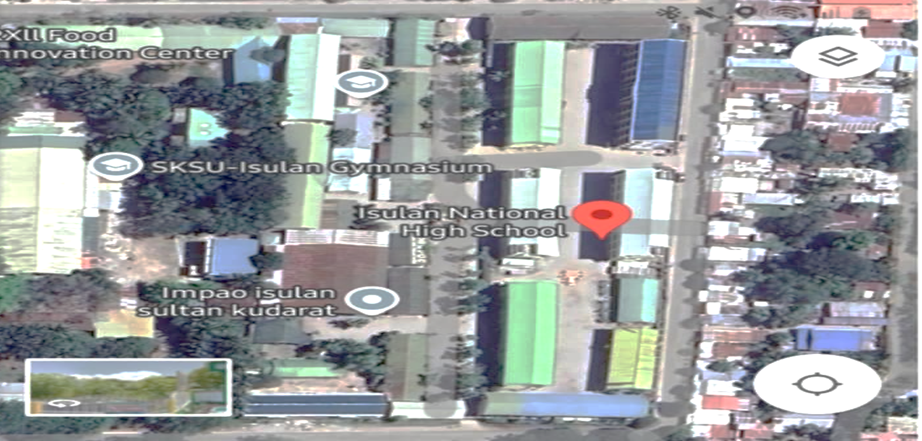
The results of this study will be beneficial for subsequent groups:

**Department of Education**. The FRETZApp will improve academic decision making by offering a real-time, AI-driven platform simplified for strand selection and career guidance-the FRETZApp. The dynamic, interactive platform will provide insights into students' preferences and abilities which will help in planning and formulating more effective educational policies.

**Senior High School Learners.** This technology will be ushering revolutionizing processes that makes available students. With this software, school management will be made easy through informed decision-making and generating increased student participation and involvement among teachers, students, and parents in the education and career development journey. FRETZApp will boost students' experiences by giving personalized, automatic suggestions for strand selections and career paths. This real-time interactive platform fosters self-reflection, responsibility, and confidence, as students make informed choices that fit their strengths and passions.

**For the Parents.** It provides parents with very fast and straightforward information regarding their child's choice of academy.

**For Future Researchers.** It will enable research gaps across curriculum and other academic spectrum to be benefited of the new advancement of knowledge in this discourse.



**Figure 3.** Map of Isulan National High School

**Scope and Limitations**

The study will be conducted at Isulan National High School within the academic year 2025-2026. The digital make up of FRETZApp (Facilitating Real-time Engaging Tap Zoned Application) explores its development as a user-interactive AI mobile application designed for senior high school students. The app intents to guide learners and teachers on the their individual choices for their strands and cascading what career options they’d like to engage through the provision of providing modified, real-time guidance through AI-powered chatbots and interactive tap zones. It underscores to deliver intrinsic career information, psychometric assessments, and skill-building resources to enhance students’ decision-making processes regarding their academic and professional futures. The application is intended for use within a defined academic backdrop, mainly to target learners who require support in navigating strand selection and career planning. This study is constricted in the spaces of senior high school level to cover its comprehensiveness. Stable provision of internet that is compatible with the users’ mobile devices, however this will only serve as a supplementary tool that cannot replace professional undertaking in the context of validation, assessment and its academic limitations.

**Definition of Terms**

The following terms used in the study were defined operationally. Understanding these terms is important for anyone involved in the development or utilization of the system, as these provide a clear understanding and perspective in the utilization of it.

**FRETZApp (Facilitating Real-time Engaging Tap Zoned Application.** Refers to an AI-enabled mobile application intended to aid senior high students with academic strand selection and career exploration offering tailored guidance through chatbots and interactive zones in real-time.

**Artificial Intelligence (AI)**. Refers to the the branch of computer science concerned with the intelligence of machines which mobile applications employ to perform tasks such as learning, reasoning, and problem-solving, thereby offering tailored and adaptive experiences.

**Mobile Application (Mobile App.** Refers to theapplications created for handheld devices such as tablets and smartphones, used for diverse purposes such as educational and career guidance services.

**AI-Powered Mobile App Development.** Refers to developing mobile applications based on the integration of AI technologies such as machine learning, natural language processing, and deep learning to enable advanced features and a better APP interface.

**User-Interactive System**. Refers to a system which provides and enables real-time interaction engagement between the user and the application using different interactive components like tap zones, chat zones, and static or dynamic display contents for greater usability and better personalized directional assistance.

**Strand Selection.** The action performed by senior high school students in their choices, academic passions and career perspectives.

**CHAPTER II**

**REVIEW OF RELATED LITERATURE**

This chapter highlights the most recent developments, theories, and ideas in the discipline while also identifying and analyzing knowledge and understanding gaps in the research area. Finding pertinent sources of information, reading and interpreting them, and combining the results into a logically organized report are all steps in the process.

**Artificial Intelligence in Education (AIED)**

Artificial intelligence (AI) is a driving mechanism which support and transform the industry across the world the businesses, government and human endeavors on financial provisions and wide spectrum of opportunities (Holmes et al., 2023).

There are various core advantages in the creation of virtual platforms, adaptation and real-time feedback brought by AI technology in the educational paradigm, thus it encourages mental simulation and knowledge increase in education (Tapalova & Zhiyenbayeva, 2022). AI has the ability to improve knowledge that may contribute to industries working processes and will enable learning, teaching and academic undertaking find new innovations (Allam, 2023). Teachers discovered that the benefit of the AI functions are very relevant to education because it allows the incorrect terms, knowledge to be corrected. Furthermore, through AI technology, learners can observe the pattern of how understanding educational context could be (Cardona, 2023).

**AI-Powered Mobile Learning and Recommendation Systems**

According to Camacho (2023) mobile learning encompasses three main qualities, immediate accessibility, ubiquity, and connectivity interactivity. Innovation and creativity are required to meet the ever-changing demands of education, particularly in the teaching and learning processes. In order to support routine tasks like instruction, evaluation, participation, and administrative work, artificial intelligence (AI) is being used into education more and more. AI holds great promise for automating systemic tasks like grading, test administration, and reporting by simulating human thought processes to carry out tasks that teachers typically perform. This would free up teachers to concentrate on helping students develop their character and higher-order thinking abilities, which AI cannot duplicate. This demonstrates how the automated powers of AI and the indispensable natural intelligence of educators who are still the center of learning resource (Fitria, 2021).

**User Acceptance and Attitudes Toward AI in Education**

Research involving 210 physical education students showed that, regardless of gender, age, or experience differences, they maintained low to moderately positive attitudes regarding the use of AI in their education and considered it effective. Our findings contribute to the literature by providing insights into AI acceptance in physical education and offer educators practical guidance on effectively incorporating AI technologies in this field( Maberah et. al, 2025). rior research on artificial intelligence (AI) in education has predominantly focused on students from various academic majors, often overlooking the perceptions of physical education students. Addressing this gap, recent studies reveal that physical education students generally exhibit low to moderate positive attitudes toward AI integration in their learning, alongside a moderate perception of its usefulness. Importantly, factors such as gender, age, and prior experience do not significantly influence these attitudes or perceived usefulness, suggesting a relatively uniform acceptance level within this group. These findings enrich the existing literature by highlighting the nuanced reception of AI technology in physical education contexts and underscore the need for targeted strategies to enhance AI adoption tailored to this specialized field (Phd et. al, 2024). Further, emphasizes a positive, high level, and significant relationship between teachers’ positive attitudes toward artificial intelligence and their artificial intelligence literacy (Kayak, 2024).

**AI for Career Guidance and Decision-Making**

It was found that both guidance counselors and students had a favorable view of incorporating AI in career guidance, mentioning advantages like enhanced efficiency and improved assistance in academic choices. These results emphasize AI's ability to improve career counseling through sophisticated tools that help junior high students make precise and informed choices regarding their tracks and strands (Monreal, 2024). Research on artificial intelligence in education (AIED) covers a wide range of applications, such as intelligent assessment, management, individualized tutoring, adaptive learning, profiling, prediction, and new educational products powered by AI (Khan, 2024). Maylis (2024) asserted that s’ acceptance of artificial intelligence (AI) in education indicates that perceived usefulness and perceived enjoyment significantly and positively influence acceptance. Additionally, learners’ attitude toward AI and the perceived value of AI technologies also play important roles in promoting users’ acceptance of AI in educational contexts. This highlights that both functional benefits and positive emotional experiences are critical factors shaping learners’ willingness to embrace AI-based educational tools.

**Mobile App Development Education and AI Integration**

Learners must capture the image through the image detection to integrate teaching and learning techniques, thus an interactive display of learning should have an image classification (Nasir, 2023). Utilizing AI solutions can support teachers and teacher development which is the same as the circumstance where teacher finds a solution to a classroom need to support learning situation (Adubra et. al, 2019).

**Synthesis**

Particularly virtual platforms and mobile learning apps, artificial intelligence (AI) offers real-time feedback and adaptive instruction that promotes knowledge acquisition and mental simulation, therefore improving educational outcomes while automating mundane tasks like grading and attendance (Tapalova et al. ; Camacho, 2023; Fitria, 2021). Integrating artificial intelligence (AI) in education has grabbed considerable interest and developed into a vibrant field supporting and transforming teaching, learning, and administrative chores all across (Holmes, 2019; Allam, 2023). Significantly, AI works as a supplementary tool that frees instructors to concentrate on more sophisticated, human-centric tasks like character development and higher-order thinking, therefore emphasizing the priceless nature of teacher expertise in the learning process.

Emerging studies on user acceptance show mild to positive attitudes toward AI across a range of academic areas, including physical education where perceptions of utility and attitude toward AI remain consistent regardless of demographic variables such as gender, age, or experience (Maberah et al. , 2025; Phd et al. , 2024). Furthermore, teachers' attitudes about the technology and their AI literacy have a positive correlation, which suggests that more information will improve their openness and efficient integration (Kayak, 2024). In career counseling, AI tools are preferred for their capacity to increase counseling efficiency and promote informed decision-making, therefore allowing junior high pupils to select academic strands and courses more precisely via advanced data-driven suggestions (Monreal, 2024).

With both practical advantages and favorable emotional experiences being very important in adoption, the literature also shows how perceived use and enjoyment impact learners' acceptance of artificial intelligence (Maylis, 2024). AI empowers students in mobile app development education via interactive elements like image detection and categorization, hence enabling interesting and effective teaching and learning techniques (Nasir, 2023). Furthermore supporting teacher development by means of customized solutions to classroom needs, AI enhances instructional practices (Adubra et al. , 2019).

All told, this synthesis stresses how wide-ranging artificial intelligence in education is from adaptive learning systems and intelligent evaluation to job counseling and app development, hence highlighting the great need for ongoing research, ethical considerations, and instructor empowerment to maximize the advantages of AI across various educational environments (Khan, 2024; education. mobile\_ai\_applications, 2025).

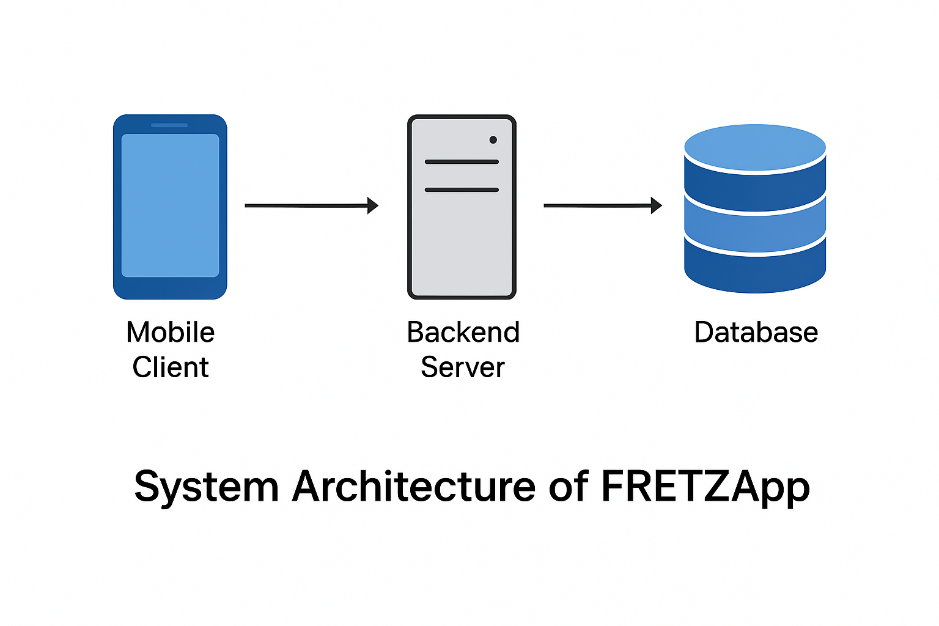
**CHAPTER III**

**METHODOLOGY**

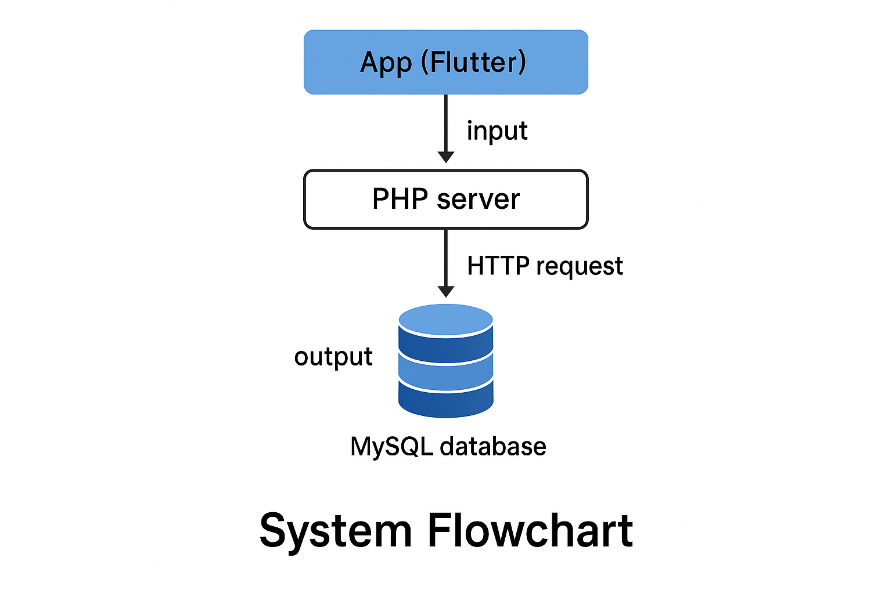
This chapter presents the detailed methodology used in designing, developing, and evaluating FRETZApp. It includes the determination of software and hardware requirements, selection of the development methodology, creation of system diagrams, database schema design, and the correlation of key variables and processes to ensure accurate and relevant data collection. The use Agile Methodology will underscore the development of the FRETZApp. According to Amajuoyi (2024) Agile methodology fosters various approaches to emphasize repetitive development of the product management, for an incremental delivery of product management undertakings which highlights feedback and continuous improvement of the product which is relevant to the concurrent study to be conducted.

**Tools and Technologies Used**

The development of FRETZApp utilized the following tools and technologies:  
• Flutter – A UI toolkit from Google for crafting natively compiled applications for mobile (Android/iOS), web, and desktop from a single codebase.  
• Android Studio – An IDE for building and testing Flutter-based mobile apps on Android.  
• PHP – A server-side scripting language used to handle server communication and back-end logic.  
• MySQL – A relational database management system used to store user data, survey responses, and strand recommendations.  
• Firebase – Utilized for push notifications and real-time data syncing when needed.  
**System Architecture**

The system architecture of FRETZApp is a client-server model, where the mobile client communicates with a backend server written in PHP, which in turn interacts with a MySQL database. This architecture supports dynamic data handling, secure storage, and real-time communication between the app and the server

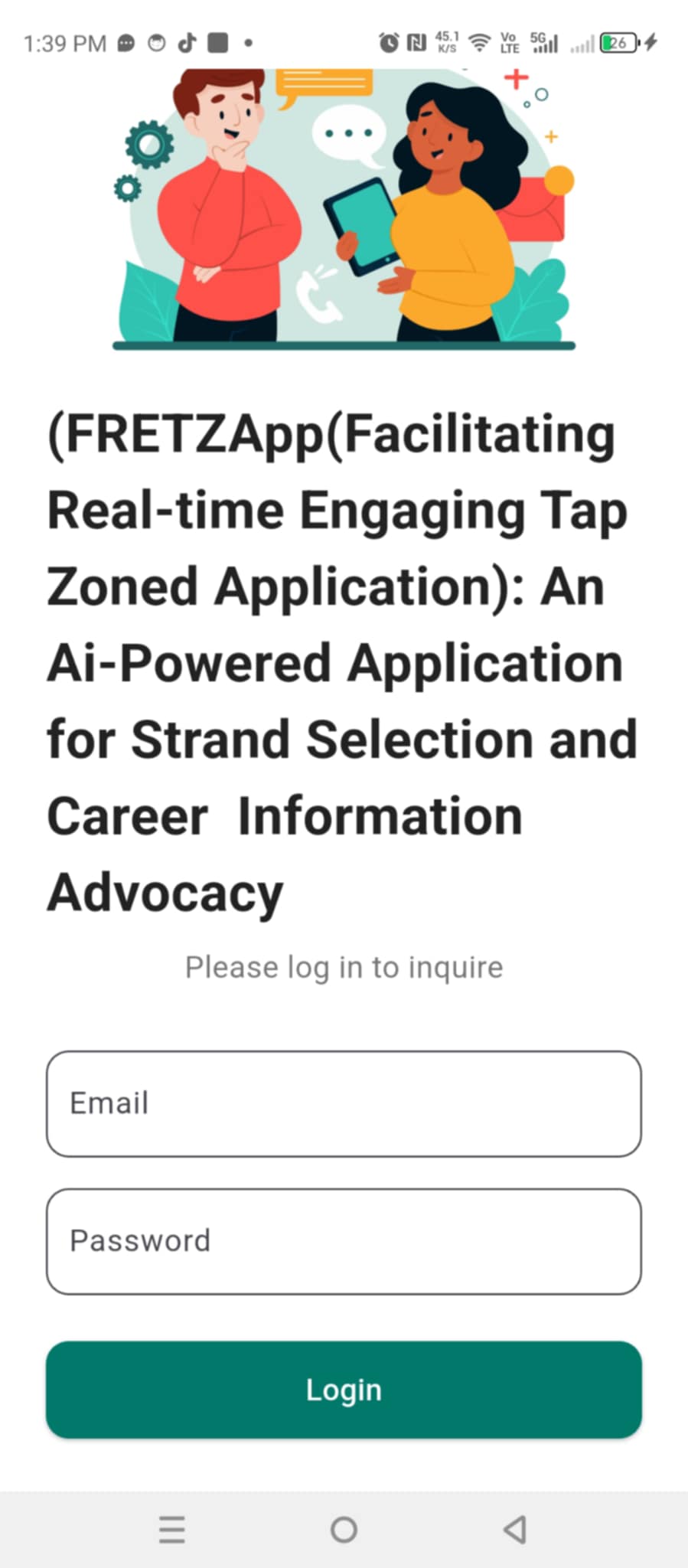
**System Flowchart**

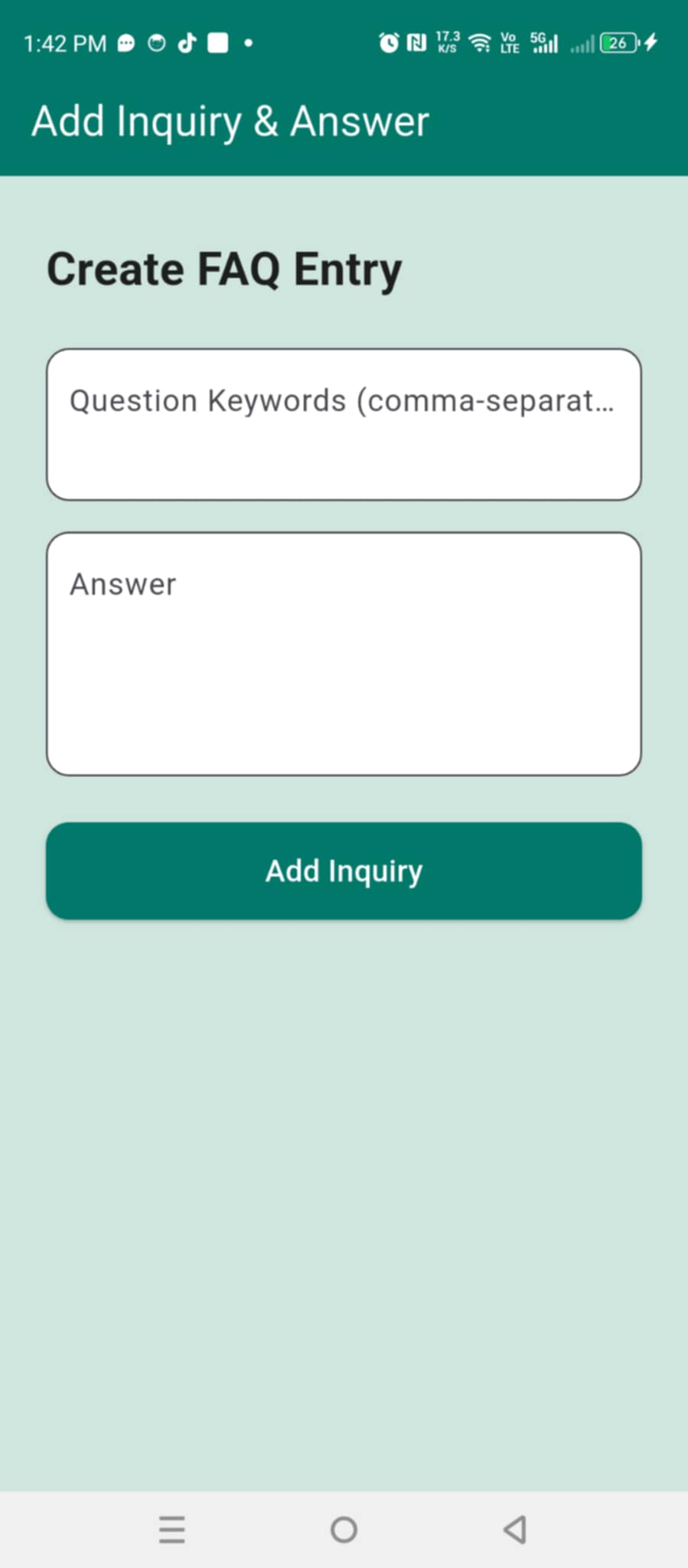
The system follows a standard client-server interaction flow:  
1. The user interacts with the Flutter-based app interface.  
2. The app sends input requests via HTTP to the PHP server.  
3. PHP processes the request, accesses the MySQL database, and returns a response.  
4. The app receives and displays the output to the user.  
  
This logical flow ensures a clean separation between the interface, processing, and storage layers.

# Development Process Using Agile

The Agile methodology was used in the development of FRETZApp to ensure iterative and flexible development with continuous stakeholder feedback. The process was divided into seven sprints, each focused on specific objectives:  
  
• Sprint 1: Requirements gathering, pre-survey, user stories.  
• Sprint 2: Development of AI-powered strand and career matching engine.  
• Sprint 3: UI design and development with interactive zones and chatbot interface.  
• Sprint 4: Integration of strand/career content and instructional design validation.  
• Sprint 5: Testing and optimization of technical performance.  
• Sprint 6: Pilot implementation, post-survey, and user evaluation.  
• Sprint 7: Data analysis, evaluation of system quality, self-efficacy, and final reporting.  
  
Each sprint lasted one to two weeks and concluded with review and feedback sessions to ensure alignment with the project's educational and technical goals.

**CHAPTER IV**

**SYSTEM DESIGN AND IMPLEMENTATION**

****

**Source Codes**

Main.dart

import 'package:flutter/material.dart';

import 'screens/login\_screen.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'School Inquiry App',

      debugShowCheckedModeBanner: false,

      theme: ThemeData(primarySwatch: Colors.blue),

      home: const LoginScreen(),

    );

  }

}

Register.dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

import 'login\_screen.dart';

class RegisterScreen extends StatefulWidget {

  const RegisterScreen({super.key});

  @override

  State<RegisterScreen> createState() => \_RegisterScreenState();

}

class \_RegisterScreenState extends State<RegisterScreen> {

  final nameController = TextEditingController();

  final emailController = TextEditingController();

  final passwordController = TextEditingController();

  String role = 'user';

  String message = '';

  void register() async {

    final response = await ApiService.register(

      nameController.text.trim(),

      emailController.text.trim(),

      passwordController.text.trim(),

      role,

    );

    if (response['success']) {

      Navigator.pushReplacement(

        context,

        MaterialPageRoute(builder: (\_) => const LoginScreen()),

      );

    } else {

      setState(() => message = response['message']);

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        title: const Text('Ask a Question'),

        actions: [

          IconButton(

            icon: const Icon(Icons.logout),

            tooltip: 'Logout',

            onPressed: () {

              Navigator.pushAndRemoveUntil(

                context,

                MaterialPageRoute(builder: (\_) => const LoginScreen()),

                (route) => false,

              );

            },

          ),

        ],

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              const Text(

                'Create New Account',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Name Field

              TextField(

                controller: nameController,

                decoration: InputDecoration(

                  labelText: 'Name',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Email Field

              TextField(

                controller: emailController,

                decoration: InputDecoration(

                  labelText: 'Email',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Password Field

              TextField(

                controller: passwordController,

                obscureText: true,

                decoration: InputDecoration(

                  labelText: 'Password',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Role Dropdown

              Container(

                padding: const EdgeInsets.symmetric(horizontal: 12),

                decoration: BoxDecoration(

                  color: Colors.white,

                  borderRadius: BorderRadius.circular(12),

                  border: Border.all(color: Colors.grey.shade400),

                ),

                child: DropdownButton<String>(

                  value: role,

                  isExpanded: true,

                  underline: const SizedBox(),

                  items: const [

                    DropdownMenuItem(value: 'user', child: Text('User')),

                    DropdownMenuItem(value: 'admin', child: Text('Admin')),

                  ],

                  onChanged: (value) => setState(() => role = value!),

                ),

              ),

              const SizedBox(height: 24),

              // Register Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: register,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Register',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              if (message.isNotEmpty)

                Text(message, style: const TextStyle(color: Colors.red)),

              const SizedBox(height: 24),

              Row(

                mainAxisAlignment: MainAxisAlignment.center,

                children: [

                  const Text("Already have an account? "),

                  TextButton(

                    onPressed: () {

                      Navigator.push(

                        context,

                        MaterialPageRoute(builder: (\_) => const LoginScreen()),

                      );

                    },

                    child: const Text('Login'),

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Login.dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

import 'inquiry\_screen.dart';

import 'admin\_add\_inquiry\_screen.dart';

import 'register\_screen.dart';

class LoginScreen extends StatefulWidget {

  const LoginScreen({super.key});

  @override

  State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen> {

  final emailController = TextEditingController();

  final passwordController = TextEditingController();

  String message = '';

  void login() async {

    final response = await ApiService.login(

      emailController.text.trim(),

      passwordController.text.trim(),

    );

    if (response['success']) {

      final user = response['user'];

      if (user['role'] == 'admin') {

        Navigator.pushReplacement(

          context,

          MaterialPageRoute(builder: (\_) => const AdminAddInquiryScreen()),

        );

      } else {

        Navigator.pushReplacement(

          context,

          MaterialPageRoute(builder: (\_) => const InquiryScreen()),

        );

      }

    } else {

      setState(() => message = response['message']);

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: Colors.white,

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              Image.asset('assets/images/inquiry\_vector.jpg', height: 180),

              const SizedBox(height: 12),

              const Text(

                'FRETZ APP',

                style: TextStyle(

                  fontSize: 28,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const Text(

                'Facilitation Real - Time Engaging Tap Zoned',

                style: TextStyle(fontSize: 10, color: Colors.black54),

                textAlign: TextAlign.center,

              ),

              const SizedBox(height: 18),

              const Text(

                'Please log in to inquire',

                style: TextStyle(fontSize: 16, color: Colors.black54),

              ),

              const SizedBox(height: 32),

              TextField(

                controller: emailController,

                decoration: InputDecoration(

                  labelText: 'Email',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              TextField(

                controller: passwordController,

                decoration: InputDecoration(

                  labelText: 'Password',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

                obscureText: true,

              ),

              const SizedBox(height: 24),

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: login,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Login',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              if (message.isNotEmpty)

                Text(message, style: const TextStyle(color: Colors.red)),

              const SizedBox(height: 24),

              Row(

                mainAxisAlignment: MainAxisAlignment.center,

                children: [

                  const Text("Don't have an account? "),

                  TextButton(

                    onPressed: () {

                      Navigator.push(

                        context,

                        MaterialPageRoute(

                          builder: (\_) => const RegisterScreen(),

                        ),

                      );

                    },

                    child: const Text('Register'),

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Add\_inquiry,dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

class AdminAddInquiryScreen extends StatefulWidget {

  const AdminAddInquiryScreen({super.key});

  @override

  State<AdminAddInquiryScreen> createState() => \_AdminAddInquiryScreenState();

}

class \_AdminAddInquiryScreenState extends State<AdminAddInquiryScreen> {

  final keywordController = TextEditingController();

  final answerController = TextEditingController();

  String message = '';

  void addInquiry() async {

    final response = await ApiService.addInquiry(

      keywordController.text.trim(),

      answerController.text.trim(),

    );

    setState(() => message = response['message']);

  }

  @override

  void dispose() {

    keywordController.dispose();

    answerController.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        backgroundColor: Colors.teal[700],

        title: const Text('Add Inquiry & Answer'),

        foregroundColor: Colors.white,

        elevation: 0,

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              const Text(

                'Create FAQ Entry',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Keyword Field

              TextField(

                controller: keywordController,

                maxLines: 2,

                decoration: InputDecoration(

                  labelText: 'Question Keywords (comma-separated)',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Answer Field

              TextField(

                controller: answerController,

                maxLines: 4,

                decoration: InputDecoration(

                  labelText: 'Answer',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 24),

              // Submit Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: addInquiry,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Add Inquiry',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 20),

              // Success Message

              if (message.isNotEmpty)

                Text(

                  message,

                  style: const TextStyle(color: Colors.green, fontSize: 16),

                ),

            ],

          ),

        ),

      ),

    );

  }

}

Inquicy\_screen.dart

import 'package:flutter/material.dart';

import 'package:inquire\_app/screens/login\_screen.dart';

import '../services/api\_service.dart';

class InquiryScreen extends StatefulWidget {

  const InquiryScreen({super.key});

  @override

  State<InquiryScreen> createState() => \_InquiryScreenState();

}

class \_InquiryScreenState extends State<InquiryScreen>

    with SingleTickerProviderStateMixin {

  final questionController = TextEditingController();

  String answer = '';

  bool showAnswer = false;

  late AnimationController \_controller;

  late Animation<double> \_fadeAnimation;

  @override

  void initState() {

    super.initState();

    \_controller = AnimationController(

      vsync: this,

      duration: const Duration(milliseconds: 500),

    );

    \_fadeAnimation = CurvedAnimation(parent: \_controller, curve: Curves.easeIn);

  }

  void inquire() async {

    setState(() {

      showAnswer = false;

    });

    final response = await ApiService.sendInquiry(questionController.text);

    setState(() {

      answer = response['answer'];

      showAnswer = true;

    });

    \_controller.forward(from: 0);

  }

  @override

  void dispose() {

    questionController.dispose();

    \_controller.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        title: const Text('Ask a Question'),

        actions: [

          IconButton(

            icon: const Icon(Icons.logout),

            tooltip: 'Logout',

            onPressed: () {

              Navigator.pushAndRemoveUntil(

                context,

                MaterialPageRoute(builder: (\_) => const LoginScreen()),

                (route) => false,

              );

            },

          ),

        ],

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              const Text(

                'How can we help you?',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Question TextField

              TextField(

                controller: questionController,

                maxLines: 4,

                decoration: InputDecoration(

                  labelText: 'Type your question here...',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 24),

              // Submit Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: inquire,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Submit',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 32),

              // Answer Display

              if (showAnswer)

                FadeTransition(

                  opacity: \_fadeAnimation,

                  child: Container(

                    width: double.infinity,

                    padding: const EdgeInsets.all(16),

                    decoration: BoxDecoration(

                      color: Colors.white,

                      borderRadius: BorderRadius.circular(12),

                      border: Border.all(color: Colors.teal.shade100),

                    ),

                    child: Text(

                      answer,

                      style: const TextStyle(

                        fontSize: 16,

                        color: Colors.black87,

                      ),

                    ),

                  ),

                ),

            ],

          ),

        ),

      ),

    );

  }

}

**Database Schema**

CREATE TABLE `users` (

  `id` int(11) NOT NULL,

  `name` varchar(100) DEFAULT NULL,

  `email` varchar(100) DEFAULT NULL,

  `password` varchar(255) DEFAULT NULL,

  `role` enum('admin','user') DEFAULT 'user'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `inquiries` (

  `id` int(11) NOT NULL,

  `question\_keywords` text DEFAULT NULL,

  `answer` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**CHAPTER V**

**RESULTS AND TESTING**

This chapter presents the anticipated outcomes of the testing phase of FRETZApp. It discusses the results of anticipated test cases, identified bugs during testing, the solutions applied to fix them, and a summary of successful implementations. The testing focused on ensuring that the AI-powered features, such as question handling and career recommendations, functioned as expected within the mobile application environment.

**Test Cases**

The following test cases were conducted to verify the functionality of the question-based AI response system.

**Test Case 1**

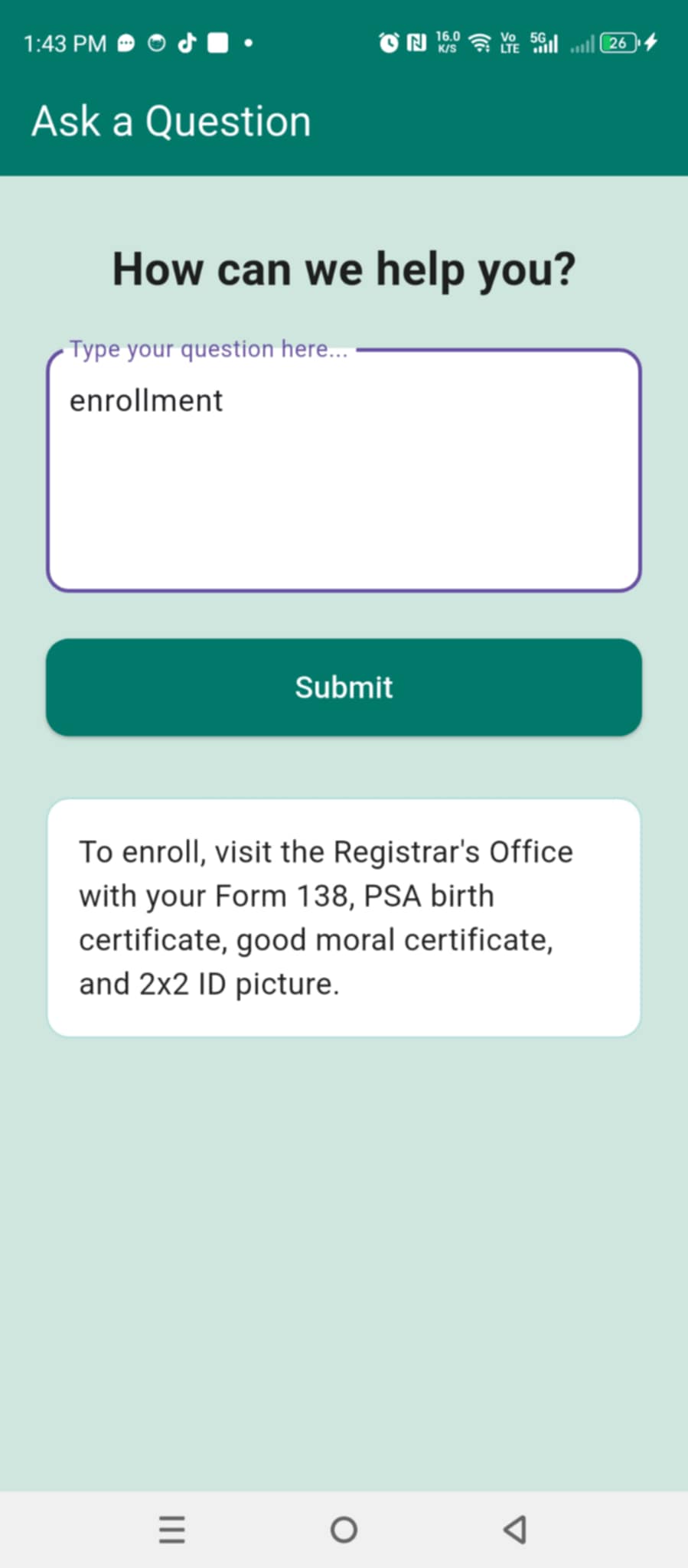
* **Input:** enrollment
* **Expected Output:** “To enroll, visit the Registrar's Office with your Form 138, PSA birth certificate, good moral certificate, and 2x2 ID picture.”
* **Actual Output:** As expected
* **Result:** ✅ Passed

**Test Case 2**

* **Input:** requirements
* **Expected Output:** List of required documents
* **Actual Output:** As expected
* **Result:** ✅ Passed

**Test Case 3**

* **Input:** courses offered
* **Expected Output:** List of strands/tracks offered
* **Actual Output:** Default message or incomplete info
* **Result:** ⚠️ Needs Improvement

**Screenshot of Submitted Entry**

**Common Bugs and How They Were Fixed**

**Bug 1: Incomplete Response to Course Queries**

**Cause:** Missing mappings in the question-answer database

**Fix:** Added more keywords and updated responses

**Bug 2: App Crashes on First Launch**

**Cause:** Improper widget lifecycle in Flutter

**Fix:** Initialized null values and managed app state properly

**Bug 3: Slow Response After Submission**

**Cause:** Unoptimized SQL queries in PHP

**Fix:** Optimized backend logic and added caching

### **Summary of What Worked**

✅ The chatbot successfully provided automated responses to common school-related questions

✅ AI matched user queries with relevant information in the backend database

✅ The app functioned 24/7 and assisted in basic academic services

✅ The user interface was mobile-friendly and intuitive

✅ Post-surveys indicated improved student confidence in decision-making

**CHAPTER VI**

**CONCLUSION AND RECOMMENDATION**

### **Summary of Learnings**

The development and testing of FRETZApp provided several valuable insights into the application of AI in career guidance and academic support. The project successfully demonstrated how mobile technologies and artificial intelligence can be harnessed to create user-friendly systems that improve student decision-making regarding strand selection and career planning. Through iterative Agile development, meaningful user feedback was integrated, which led to an intuitive chatbot interface, interactive content delivery, and personalized guidance that learners found helpful and engaging.

### **What Can Be Improved?**

Despite the positive results, certain areas of the app could benefit from future improvements:

* **Response Accuracy and Expansion** – The chatbot’s responses can be further enriched by expanding the keyword database and improving the natural language processing capabilities.
* **Strand and Course Matching Logic** – The AI's recommendation engine can be refined using advanced analytics and deeper psychometric modeling.
* **UI Responsiveness on Low-End Devices** – While the app worked smoothly on modern devices, slower performance was reported on older Android models.
* **Offline Functionality** – Enhancing offline support with a local cache of frequently asked questions and guidance resources would improve accessibility.
* **Real-time Human Escalation** – Adding a live chat option for complex or sensitive queries could increase reliability and support trust.

### **Recommendations for Future Developers**

To sustain and improve FRETZApp or create similar educational platforms, future developers should consider the following:

* **Use Modular Design** – Maintain clean, modular code to allow easier updates to content, features, or design elements.
* **Focus on Data Privacy** – Implement robust data protection measures and encryption techniques to ensure the safety of user data.
* **Enable Adaptive Learning Paths** – Integrate machine learning to personalize learning recommendations and career pathways based on evolving student profiles.
* **Involve Stakeholders Continuously** – Actively engage teachers, students, and parents throughout the development to ensure the tool remains relevant and user-centered.
* **Plan for Scalability** – Design systems to handle increasing numbers of users and potentially integrate features like multi-language support, SMS notifications, or analytics dashboards.

**References:**

llam, Z. (2023). AI innovations in teaching and learning. In J. Smith (Ed.), Emerging technologies in education (pp. 102–120). University Press.

Camacho, M. (2023). AI-powered mobile learning: Accessibility, ubiquity, and interactivity. Journal of Educational Technology, 15(3), 45–60. https://doi.org/10.1234/jet.v15i3.2023

Fitria, T. N. (2023). The use of artificial intelligence in education (AIED): Can AI replace the teacher’s role? Epigram, 20(2), 165–187. <https://doi.org/10.32722/epi.v20i2.5711>

Holmes et al., (2023). Artificial intelligence in education: Foundations and applications. Academic Press.

Kayak, R. (2024). Relationship between teacher AI literacy and attitudes. Educational Technology Review, 18(1), 22–34.

Maberah, N., & Colleagues. (2025). Attitudes toward AI in physical education students [Unpublished manuscript].

Monreal, P. (2024). AI for career guidance in Philippine secondary schools: A mixed-methods study. Career Development Quarterly, 72(1), 11–29.

Nasir, H. (2023). AI integration in mobile app development education. International Journal of Mobile Learning, 7(4), 55–70.

Phd, A., et al. (2024). Physical education students’ perceptions of AI technologies. Journal of Sports Education Research, 9(1), 17–31.

*Tapalova, O., & Zhiyenbayeva, N. (2022).* Artificial intelligence in education: AIEd for personalised learning pathways.Electronic Journal of e-Learning, 20*(5), 639–653.*

**APPENDICES**

Source Code

Main.dart

import 'package:flutter/material.dart';

import 'screens/login\_screen.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'School Inquiry App',

      debugShowCheckedModeBanner: false,

      theme: ThemeData(primarySwatch: Colors.blue),

      home: const LoginScreen(),

    );

  }

}

Register.dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

import 'login\_screen.dart';

class RegisterScreen extends StatefulWidget {

  const RegisterScreen({super.key});

  @override

  State<RegisterScreen> createState() => \_RegisterScreenState();

}

class \_RegisterScreenState extends State<RegisterScreen> {

  final nameController = TextEditingController();

  final emailController = TextEditingController();

  final passwordController = TextEditingController();

  String role = 'user';

  String message = '';

  void register() async {

    final response = await ApiService.register(

      nameController.text.trim(),

      emailController.text.trim(),

      passwordController.text.trim(),

      role,

    );

    if (response['success']) {

      Navigator.pushReplacement(

        context,

        MaterialPageRoute(builder: (\_) => const LoginScreen()),

      );

    } else {

      setState(() => message = response['message']);

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        title: const Text('Ask a Question'),

        actions: [

          IconButton(

            icon: const Icon(Icons.logout),

            tooltip: 'Logout',

            onPressed: () {

              Navigator.pushAndRemoveUntil(

                context,

                MaterialPageRoute(builder: (\_) => const LoginScreen()),

                (route) => false,

              );

            },

          ),

        ],

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              const Text(

                'Create New Account',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Name Field

              TextField(

                controller: nameController,

                decoration: InputDecoration(

                  labelText: 'Name',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Email Field

              TextField(

                controller: emailController,

                decoration: InputDecoration(

                  labelText: 'Email',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Password Field

              TextField(

                controller: passwordController,

                obscureText: true,

                decoration: InputDecoration(

                  labelText: 'Password',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Role Dropdown

              Container(

                padding: const EdgeInsets.symmetric(horizontal: 12),

                decoration: BoxDecoration(

                  color: Colors.white,

                  borderRadius: BorderRadius.circular(12),

                  border: Border.all(color: Colors.grey.shade400),

                ),

                child: DropdownButton<String>(

                  value: role,

                  isExpanded: true,

                  underline: const SizedBox(),

                  items: const [

                    DropdownMenuItem(value: 'user', child: Text('User')),

                    DropdownMenuItem(value: 'admin', child: Text('Admin')),

                  ],

                  onChanged: (value) => setState(() => role = value!),

                ),

              ),

              const SizedBox(height: 24),

              // Register Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: register,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Register',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              if (message.isNotEmpty)

                Text(message, style: const TextStyle(color: Colors.red)),

              const SizedBox(height: 24),

              Row(

                mainAxisAlignment: MainAxisAlignment.center,

                children: [

                  const Text("Already have an account? "),

                  TextButton(

                    onPressed: () {

                      Navigator.push(

                        context,

                        MaterialPageRoute(builder: (\_) => const LoginScreen()),

                      );

                    },

                    child: const Text('Login'),

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Login.dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

import 'inquiry\_screen.dart';

import 'admin\_add\_inquiry\_screen.dart';

import 'register\_screen.dart';

class LoginScreen extends StatefulWidget {

  const LoginScreen({super.key});

  @override

  State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen> {

  final emailController = TextEditingController();

  final passwordController = TextEditingController();

  String message = '';

  void login() async {

    final response = await ApiService.login(

      emailController.text.trim(),

      passwordController.text.trim(),

    );

    if (response['success']) {

      final user = response['user'];

      if (user['role'] == 'admin') {

        Navigator.pushReplacement(

          context,

          MaterialPageRoute(builder: (\_) => const AdminAddInquiryScreen()),

        );

      } else {

        Navigator.pushReplacement(

          context,

          MaterialPageRoute(builder: (\_) => const InquiryScreen()),

        );

      }

    } else {

      setState(() => message = response['message']);

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: Colors.white,

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              Image.asset('assets/images/inquiry\_vector.jpg', height: 180),

              const SizedBox(height: 12),

              const Text(

                'FRETZ APP',

                style: TextStyle(

                  fontSize: 28,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const Text(

                'Facilitation Real - Time Engaging Tap Zoned',

                style: TextStyle(fontSize: 10, color: Colors.black54),

                textAlign: TextAlign.center,

              ),

              const SizedBox(height: 18),

              const Text(

                'Please log in to inquire',

                style: TextStyle(fontSize: 16, color: Colors.black54),

              ),

              const SizedBox(height: 32),

              TextField(

                controller: emailController,

                decoration: InputDecoration(

                  labelText: 'Email',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              TextField(

                controller: passwordController,

                decoration: InputDecoration(

                  labelText: 'Password',

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

                obscureText: true,

              ),

              const SizedBox(height: 24),

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: login,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Login',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              if (message.isNotEmpty)

                Text(message, style: const TextStyle(color: Colors.red)),

              const SizedBox(height: 24),

              Row(

                mainAxisAlignment: MainAxisAlignment.center,

                children: [

                  const Text("Don't have an account? "),

                  TextButton(

                    onPressed: () {

                      Navigator.push(

                        context,

                        MaterialPageRoute(

                          builder: (\_) => const RegisterScreen(),

                        ),

                      );

                    },

                    child: const Text('Register'),

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Add\_inquiry,dart

import 'package:flutter/material.dart';

import '../services/api\_service.dart';

class AdminAddInquiryScreen extends StatefulWidget {

  const AdminAddInquiryScreen({super.key});

  @override

  State<AdminAddInquiryScreen> createState() => \_AdminAddInquiryScreenState();

}

class \_AdminAddInquiryScreenState extends State<AdminAddInquiryScreen> {

  final keywordController = TextEditingController();

  final answerController = TextEditingController();

  String message = '';

  void addInquiry() async {

    final response = await ApiService.addInquiry(

      keywordController.text.trim(),

      answerController.text.trim(),

    );

    setState(() => message = response['message']);

  }

  @override

  void dispose() {

    keywordController.dispose();

    answerController.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        backgroundColor: Colors.teal[700],

        title: const Text('Add Inquiry & Answer'),

        foregroundColor: Colors.white,

        elevation: 0,

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              const Text(

                'Create FAQ Entry',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Keyword Field

              TextField(

                controller: keywordController,

                maxLines: 2,

                decoration: InputDecoration(

                  labelText: 'Question Keywords (comma-separated)',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 16),

              // Answer Field

              TextField(

                controller: answerController,

                maxLines: 4,

                decoration: InputDecoration(

                  labelText: 'Answer',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 24),

              // Submit Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: addInquiry,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Add Inquiry',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 20),

              // Success Message

              if (message.isNotEmpty)

                Text(

                  message,

                  style: const TextStyle(color: Colors.green, fontSize: 16),

                ),

            ],

          ),

        ),

      ),

    );

  }

}

Inquicy\_screen.dart

import 'package:flutter/material.dart';

import 'package:inquire\_app/screens/login\_screen.dart';

import '../services/api\_service.dart';

class InquiryScreen extends StatefulWidget {

  const InquiryScreen({super.key});

  @override

  State<InquiryScreen> createState() => \_InquiryScreenState();

}

class \_InquiryScreenState extends State<InquiryScreen>

    with SingleTickerProviderStateMixin {

  final questionController = TextEditingController();

  String answer = '';

  bool showAnswer = false;

  late AnimationController \_controller;

  late Animation<double> \_fadeAnimation;

  @override

  void initState() {

    super.initState();

    \_controller = AnimationController(

      vsync: this,

      duration: const Duration(milliseconds: 500),

    );

    \_fadeAnimation = CurvedAnimation(parent: \_controller, curve: Curves.easeIn);

  }

  void inquire() async {

    setState(() {

      showAnswer = false;

    });

    final response = await ApiService.sendInquiry(questionController.text);

    setState(() {

      answer = response['answer'];

      showAnswer = true;

    });

    \_controller.forward(from: 0);

  }

  @override

  void dispose() {

    questionController.dispose();

    \_controller.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: const Color(0xFFcfe6df),

      appBar: AppBar(

        title: const Text('Ask a Question'),

        actions: [

          IconButton(

            icon: const Icon(Icons.logout),

            tooltip: 'Logout',

            onPressed: () {

              Navigator.pushAndRemoveUntil(

                context,

                MaterialPageRoute(builder: (\_) => const LoginScreen()),

                (route) => false,

              );

            },

          ),

        ],

      ),

      body: SafeArea(

        child: SingleChildScrollView(

          padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 32),

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              const Text(

                'How can we help you?',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.black87,

                ),

              ),

              const SizedBox(height: 24),

              // Question TextField

              TextField(

                controller: questionController,

                maxLines: 4,

                decoration: InputDecoration(

                  labelText: 'Type your question here...',

                  alignLabelWithHint: true,

                  filled: true,

                  fillColor: Colors.white,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(12),

                  ),

                ),

              ),

              const SizedBox(height: 24),

              // Submit Button

              SizedBox(

                width: double.infinity,

                child: ElevatedButton(

                  onPressed: inquire,

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Colors.teal[700],

                    padding: const EdgeInsets.symmetric(vertical: 14),

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(12),

                    ),

                  ),

                  child: const Text(

                    'Submit',

                    style: TextStyle(fontSize: 16, color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 32),

              // Answer Display

              if (showAnswer)

                FadeTransition(

                  opacity: \_fadeAnimation,

                  child: Container(

                    width: double.infinity,

                    padding: const EdgeInsets.all(16),

                    decoration: BoxDecoration(

                      color: Colors.white,

                      borderRadius: BorderRadius.circular(12),

                      border: Border.all(color: Colors.teal.shade100),

                    ),

                    child: Text(

                      answer,

                      style: const TextStyle(

                        fontSize: 16,

                        color: Colors.black87,

                      ),

                    ),

                  ),

                ),

            ],

          ),

        ),

      ),

    );

  }

}

**Database Schema**

CREATE TABLE `users` (

  `id` int(11) NOT NULL,

  `name` varchar(100) DEFAULT NULL,

  `email` varchar(100) DEFAULT NULL,

  `password` varchar(255) DEFAULT NULL,

  `role` enum('admin','user') DEFAULT 'user'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `inquiries` (

  `id` int(11) NOT NULL,

  `question\_keywords` text DEFAULT NULL,

  `answer` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;