LIGHTHOUSE

by

Nandana Pradeep (2141054)

Sandra Benny (2141058)

Under the guidance of

Dr. Arokia Paul Rajan

A Project report submitted in partial fulfillment of
the requirements for the award of degree of
Bachelor of Computer Applications of
CHRIST (Deemed to be University)

April - 2024

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled ***LightHouse*** *is a bona fide record of work done by* ***Nandana Pradeep (2141054) and Sandra Benny (2141058)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of VI Semester BCA during the year 2024.*

Head of the Department **Faculty In-charge**

Valued-by:	Name	:	Nandana Pradeep, Sandra Benny
1.	Register Numbers	:	2141054 2141058
2.	Examination center	:	CHRIST (Deemed to be University)
	Date	:	

ACKNOWLEDGMENTS

First of all, we thank God the Almighty for his immense grace and blessings showered on us at every stage of this work.

We extend our heartfelt gratitude to **Dr. Ashok Immanuel V**, the Head of the Computer Science Department at CHRIST (Deemed to be University), for his unwavering support and guidance throughout our project and to **Dr. Beaulah Soundarabai P**, the Coordinator of the Computer Science Department, for her invaluable insights and encouragement. Her dedication to fostering excellence in education has inspired us to strive for innovation and excellence in our work.

Our profound appreciation goes to our project guide, **Dr. Arokia Paul Rajan**, Associate Professor in the Computer Science Department at CHRIST (Deemed to be University). His mentorship, expertise, and unwavering support have been instrumental in steering our project towards its goals.

Lastly, we would like to extend our gratitude to all the faculty and staff members of the Computer Science Department at CHRIST (Deemed to be University) for their constant support, encouragement, and valuable input. Their collective efforts have created a conducive learning environment that has enriched our educational journey. Last but not least, we would like to thank everyone involved in the project directly or indirectly.

ABSTRACT

In the ever-evolving landscape of entrepreneurship and investment, LightHouse emerges as a beacon of innovation, heralding a new era of connectivity between startups and investors. With a commitment to seamless navigation and user-centric design, our platform revolutionizes the process of discovering and engaging with groundbreaking ideas. Through meticulously crafted pitches, startups unveil their visions to a discerning audience, with each presentation accompanied by a compatibility ratio, meticulously calculated through our proprietary algorithm. This algorithm, a testament to our commitment to precision and efficiency, enables investors to identify the most promising opportunities with unparalleled accuracy.

But LightHouse is more than just a platform for pitch presentations. It serves as a vibrant ecosystem where users can stay informed about the latest industry trends and news, ensuring that both startups and investors are equipped with the knowledge they need to make informed decisions. The dynamic startup feed acts as an information hub, constantly updating users on new opportunities and developments within their areas of interest.

In summary, LightHouse embodies the convergence of innovation, efficiency, and connectivity in the realm of startup investment. By providing a platform that seamlessly integrates pitching, communication, and payment functionalities, we empower both startups and investors to navigate the entrepreneurial landscape with confidence and clarity, propelling the realization of visionary ideas and strategic investments alike.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 BACKGROUND OF THE PROJECT

LightHouse stands as an innovative online investment platform, revolutionizing the connection between startups and investors within an effortlessly navigable environment. Within this cutting-edge ecosystem, startups unveil their groundbreaking ideas through meticulously crafted pitches, seamlessly posted on the platform. These pitches are intricately paired with a compatibility ratio, meticulously calculated through our unique algorithm. The startup feed acts as an information hub, ensuring users stay updated on industry trends and news, while simultaneously updating investors on new startups along with their respective compatibility ratios. Communication is streamlined, allowing for effortless exchanges through text, email, and scheduled meetings. Our platform goes beyond by providing integrated payment options, solidifying LightHouse as the ultimate facilitator of entrepreneurial dreams and strategic investments.

1.2 OBJECTIVE

Empower startups to efficiently showcase their ideas and connect with investors while providing a seamless investment experience through innovative technology and curated content on LightHouse.

1.3 PURPOSE, SCOPE AND APPLICABILITY

Purpose

- To facilitate efficient and effective connections between startups and investors, enabling the growth and success of innovative ventures.
- To streamline the investment process by offering a user-friendly interface, curated content, and integrated communication and payment solutions.

Scope

- LightHouse aims to provide a comprehensive online platform for startups to present their ideas and for investors to discover and engage with promising ventures.
- The platform includes features such as pitch posting, compatibility ratio calculation, industry trend updates, communication tools, and integrated payment options.

Applicability

- LightHouse applies to a wide range of industries and sectors, catering to startups seeking funding and investors looking for promising opportunities.
- It serves as a valuable tool for entrepreneurs, investors, and industry professionals seeking to engage in strategic partnerships and investments in innovative ventures.

1.4 MODULES

- User Registration and Authentication:

  This module manages the registration process for both startups and investors, ensuring secure user authentication.

- Startup Pitch Management:

  Startups can create, edit, and manage their pitches. This includes tools for crafting presentations and showcasing key information.

- Investor Dashboard:

  Investors have a personalized dashboard displaying recommended pitches based on their preferences, the latest trends, and updates from startups they follow.

- Compatibility Ratio Algorithm:

  This module incorporates a unique algorithm to calculate compatibility ratios for each startup-investor pair, facilitating personalized matching.

- Tiered Investment Model:

  The platform implements and manages the tiered investment model, offering different levels of access and benefits to cater to a diverse range of investors.

- Featured Startup Section:

  A rotating section that highlights and promotes selected startups, providing them with increased visibility.

- Industry-Specific Showcases:

  This module introduces industry-specific showcases and themes to diversify the focus beyond technology startups, ensuring representation across various sectors.

- Direct Investment Feature:

  Enables investors to directly invest in startups within the platform, streamlining the investment process.

- Real-Time Data Updates:

  Manages the integration of real-time data updates, ensuring that the information available to investors is current and accurate.

- Financial Reporting Tools:

  Provides startups with tools to present detailed financial information, and investors with comprehensive financial metrics, fostering transparency.

- Communication Hub:

  Facilitates communication between startups and investors through text, email, and scheduled meetings, fostering collaboration.

- Integrated Payment System:

  Manages integrated payment options to facilitate secure and seamless financial transactions between startups and investors.

1.5 MAJOR OUTCOME

- Increased funding opportunities for startups, fostering their growth and development.
- Enhanced investment portfolios for investors through access to a diverse range of innovative ventures.
- Accelerated innovation and economic growth by supporting the development and implementation of groundbreaking ideas.
- Streamlined investment processes, saving time and resources for both startups and investors.

1.6 HIGHLIGHT OF THE PROJECT

- **Efficient Connectivity**: LightHouse offers a seamless platform for startups to showcase their ideas and for investors to discover promising ventures, fostering efficient connections within the entrepreneurial ecosystem.
- **Innovative Compatibility Algorithm**: The unique compatibility ratio algorithm enhances the matchmaking process, providing startups and investors with tailored recommendations based on their preferences and interests.
- **Comprehensive Information Hub**: LightHouse serves as an information hub, keeping users updated on industry trends, news, and new startup opportunities, ensuring informed decision-making.
- **Streamlined Communication**: The platform facilitates effortless communication between startups and investors through various channels, including text, email, and scheduled meetings, enabling smooth exchanges and collaboration.
- **Empowering Entrepreneurial Dreams**: By providing startups with access to funding and investors with opportunities for strategic investments, LightHouse empowers entrepreneurial dreams, driving innovation and economic growth.
- **Diverse Industry Coverage**: With applicability across various industries and sectors, LightHouse caters to a diverse range of startups and investors, fostering a vibrant and inclusive entrepreneurial ecosystem.

1.7 TOOLS USED

- FlutterFlow
- Firebase

2. SYSTEM ANALYSIS AND REQUIREMENTS

This chapter describes the system requirements and its analysis. It includes hardware and software requirements as well as functional and non-functional requirements.

2.1 EXISTING SYSTEM

AngelList and Crunchbase are prominent platforms within the startup and investment landscape, each serving distinct purposes. AngelList primarily functions as a platform for startups seeking funding and investors interested in discovering and investing in early-stage companies. Through AngelList, startups can create profiles, pitch their ideas, and connect with accredited investors, while investors can explore opportunities and participate in funding rounds. Additionally, AngelList facilitates talent acquisition for startups and job seekers within the startup ecosystem. On the other hand, Crunchbase is renowned for its comprehensive database of companies, encompassing startups, private and public companies, and investors. It offers detailed information on companies, including funding history, leadership, products, and industry trends. Users can search and filter companies based on various criteria and access news articles and market insights. While both platforms contribute to the startup and investment ecosystem, AngelList emphasizes facilitating fundraising and connections, whereas Crunchbase focuses on providing comprehensive company data and market intelligence.

2.2 LIMITATIONS OF EXISTING SYSTEM

- **Limited Focus**: AngelList primarily caters to early-stage startups and investors, which may limit its usefulness for companies at later stages of development or for investors seeking opportunities in more established firms.
- **Accredited Investor Requirement**: AngelList restricts investment opportunities to accredited investors, which may exclude a significant portion of potential investors who do not meet the accreditation criteria.
- **Limited Company Coverage**: While Crunchbase provides comprehensive data on a wide range of companies, it may not capture every startup or small business, leading to gaps in information, particularly for companies that operate outside of mainstream industries or geographical regions.

- **User Experience**: While both platforms offer valuable features, their user interfaces may not always be intuitive or user-friendly, potentially hindering the user experience and making it challenging to navigate and utilize the platforms effectively.
- **Lack of Integration**: There may be a lack of seamless integration between different functionalities within the platforms, such as fundraising, talent acquisition, and market research, making it less convenient for users to access all the tools they need in one place.
- **Limited Investor Discovery**: Startups on AngelList may face challenges in getting noticed by potential investors, especially if they do not have extensive networks or connections within the platform, limiting their ability to secure funding.

2.3 PROPOSED SYSTEM

Building upon the innovative foundation of LightHouse, our proposed project aims to further distinguish the platform by addressing the limitations identified in AngelList and Crunchbase. To tackle the exclusive nature of accredited investors on AngelList, we propose the integration of a tiered investment model on LightHouse. This model will offer different levels of access and benefits, ensuring that a wider spectrum of investors, including non-accredited ones, can participate. In response to the heightened competition on AngelList, LightHouse will introduce a featured startup section, rotating and highlighting promising but lesser-known ventures, thus mitigating the challenge of standing out. Recognizing the historical tech-centric bias, LightHouse will implement compatibility ratios where investors with different interests would be matched with startups of various fields. Taking inspiration from Crunchbase's limitations, our proposed project will introduce a direct investment feature on LightHouse, allowing investors to seamlessly invest within the platform. To combat data staleness, LightHouse will implement real-time data updates and verification mechanisms, ensuring that investors have the latest and most accurate information at their fingertips.

2.4 BENEFITS OF THE PROPOSED SYSTEM

The proposed enhancements to LightHouse offer significant benefits, addressing key limitations of existing platforms. By introducing a tiered investment model, LightHouse expands investor accessibility to both accredited and non-accredited individuals, fostering inclusivity in startup funding. The featured startup section increases visibility for promising ventures, mitigating the challenge of standing out amidst the competition. Compatibility ratios improve investor-startup matching, ensuring more targeted connections. A direct investment feature streamlines the investment process, while real-time data updates enhance reliability. These enhancements position LightHouse as a more inclusive, efficient, and reliable platform, benefiting startups and investors alike..

2.5 FEATURES OF THE PROPOSED SYSTEM

- **Tiered Investment Model**: Offering different levels of access and benefits to accommodate both accredited and non-accredited investors, fostering inclusivity in startup funding.
- **Featured Startup Section**: Rotating and highlighting promising but lesser-known ventures to increase visibility and exposure, aiding startups in standing out amidst competition.
- **Compatibility Ratios**: Implementing algorithms to match investors with startups based on diverse interests and preferences, facilitating more meaningful and targeted connections.
- **Direct Investment Feature**: Allowing investors to seamlessly invest within the LightHouse platform, streamlining the investment process and eliminating the need for external transactions.
- **Real-Time Data Updates**: Implementing mechanisms for real-time data updates and verification to ensure investors have access to the latest and most accurate information about startups, enhancing reliability and informed decision-making.
- **Streamlined Communication**: Continuing to provide efficient communication channels between startups and investors, including text, email, and scheduled meetings, facilitating seamless exchanges and collaboration.
- **Integrated Payment Options**: Offering integrated payment solutions to further simplify the investment process and provide a secure mechanism for transactions within the platform.

2.6 SYSTEM REQUIREMENTS SPECIFICATION

2.6.1 User Characteristics

The users of LightHouse encompass a diverse range of characteristics, reflecting the dynamic nature of the startup and investment ecosystem. Startups utilizing LightHouse may vary in size, industry, and stage of development, from early-stage tech startups to established businesses seeking expansion. These entrepreneurs share common traits of innovation, vision, and ambition, seeking funding and strategic partnerships to propel their ventures forward. On the investor side, users span a spectrum of experience and financial capacity, including seasoned venture capitalists, angel investors, and individual enthusiasts. They share a keen interest in discovering promising opportunities, diversifying their portfolios, and contributing to the growth of innovative ventures. LightHouse caters to these varied user characteristics by providing a platform that facilitates efficient connections, fosters collaboration, and empowers entrepreneurial dreams.

2.6.2 SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

- Flutterflow
- Firebase

Hardware Requirements

- Processor - 1.9 gigahertz (GHz) x86- or x64-bit dual core processor with SSE2 instruction set
- Graphic card - Not required
- Disk Capacity - 1-2 Gb of Disk Capacity
- RAM - 2GB of Memory

2.6.3 CONSTRAINTS

Despite its innovative features, LightHouse may encounter constraints that could impact its functionality and effectiveness. One potential constraint is regulatory compliance, particularly regarding investment regulations and accreditation requirements for investors. Ensuring adherence to legal frameworks and navigating regulatory complexities is crucial to safeguarding the integrity of the platform and maintaining trust among users. Additionally, scalability could pose a challenge as LightHouse grows its user base and expands its offerings. Scaling infrastructure and resources to accommodate increased traffic and data volume while maintaining performance and security standards requires careful planning and investment

2.6.4 FUNCTIONAL REQUIREMENTS

- Startup Pitch Posting:
    - Startups can create and post pitches showcasing their ideas, products, and business models. The platform offers user-friendly tools for crafting compelling presentations.
    - Allows start ups to add images, video pitches, graphs as well as new start up ideas.

- Compatibility Ratio Algorithm:
    - An innovative algorithm calculates a compatibility ratio for each pitch, providing investors with a quantitative measure of alignment between their preferences and a startup's characteristics.

- Tiered Investment Model:
    - The platform introduces a tiered investment model to accommodate a diverse range of investors. Accredited investors enjoy exclusive benefits, while other tiers provide varying degrees of access and perks.

- Industry-Specific Showcases:
    - Recognizing the need to diversify beyond tech-focused startups, LightHouse introduces industry-specific showcases and themes to ensure equitable visibility for startups across various sectors.

- Direct Investment Feature:
    - LightHouse enables investors to directly invest within the platform, streamlining the investment process and enhancing user convenience.

- Communication Hub:
    - The platform serves as a communication hub where startups and investors can engage effortlessly through text, email, and scheduled meetings, fostering collaboration.

- Integrated Payment Options:
    - LightHouse goes beyond traditional platforms by providing integrated payment options, facilitating secure and seamless financial transactions between startups and investors.

- Global Expansion Strategies:
    - LightHouse implements strategies for global expansion, ensuring startups and investors from diverse regions can participate. This involves addressing regulatory challenges and tailoring the platform for different markets.

2.6.5 NON-FUNCTIONAL REQUIREMENTS

- **Performance**: This refers to how well the system performs under certain conditions, such as load times for pages, response times for user actions, and overall system speed. LightHouse should aim for fast loading times and responsiveness to ensure a smooth user experience.
- **Usability**: Usability concerns how easy and intuitive the system is to use for its intended users. LightHouse should have a user-friendly interface, clear navigation, and intuitive features to make it easy for startups and investors to accomplish their tasks and achieve their goals.
- **Security**: Security involves protecting the system and its data from unauthorized access, breaches, and malicious attacks. LightHouse should implement robust security measures, including encryption, authentication, access controls, and regular security audits, to safeguard sensitive information and maintain user trust.
- **Reliability**: Reliability relates to the system's ability to perform consistently and predictably over time without unexpected failures or errors. LightHouse should be reliable, with minimal downtime and disruptions, ensuring users can access the platform whenever they need it and rely on its functionality.
- **Scalability**: Scalability refers to the system's ability to handle increasing amounts of users, data, and traffic without sacrificing performance or functionality. As LightHouse grows its user base and expands its offerings, it should be able to scale its infrastructure and resources accordingly to accommodate increased demand and maintain optimal performance.
- **Regulatory Compliance**: Regulatory compliance involves adhering to legal requirements and industry standards relevant to the operation of the platform, particularly regarding investment regulations, data protection laws, and privacy regulations. LightHouse should ensure compliance with relevant laws and regulations to avoid legal issues and maintain the trust of its users.
- **Interoperability**: Interoperability refers to the system's ability to work seamlessly with other systems and platforms, allowing for data exchange and integration. LightHouse should support interoperability with external systems, such as payment gateways, communication tools, and third-party services, to enhance functionality and provide a more comprehensive user experience.

3. SYSTEM DESIGN

3.1 ER DIAGRAM

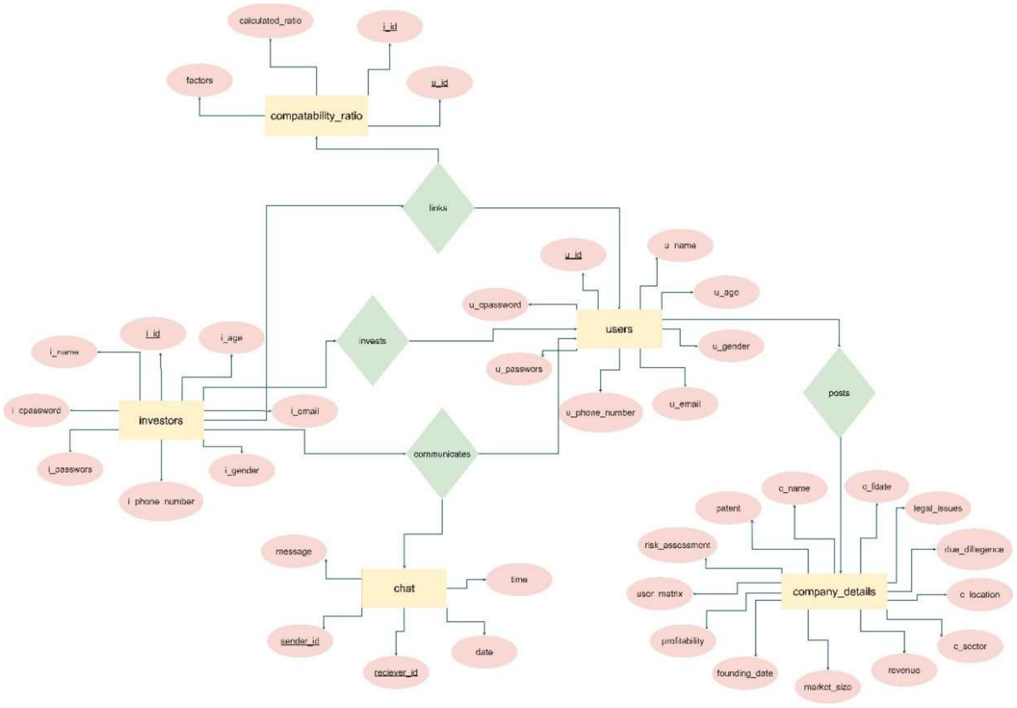


Fig 3.1 ER diagram

3.2 UML DIAGRAM

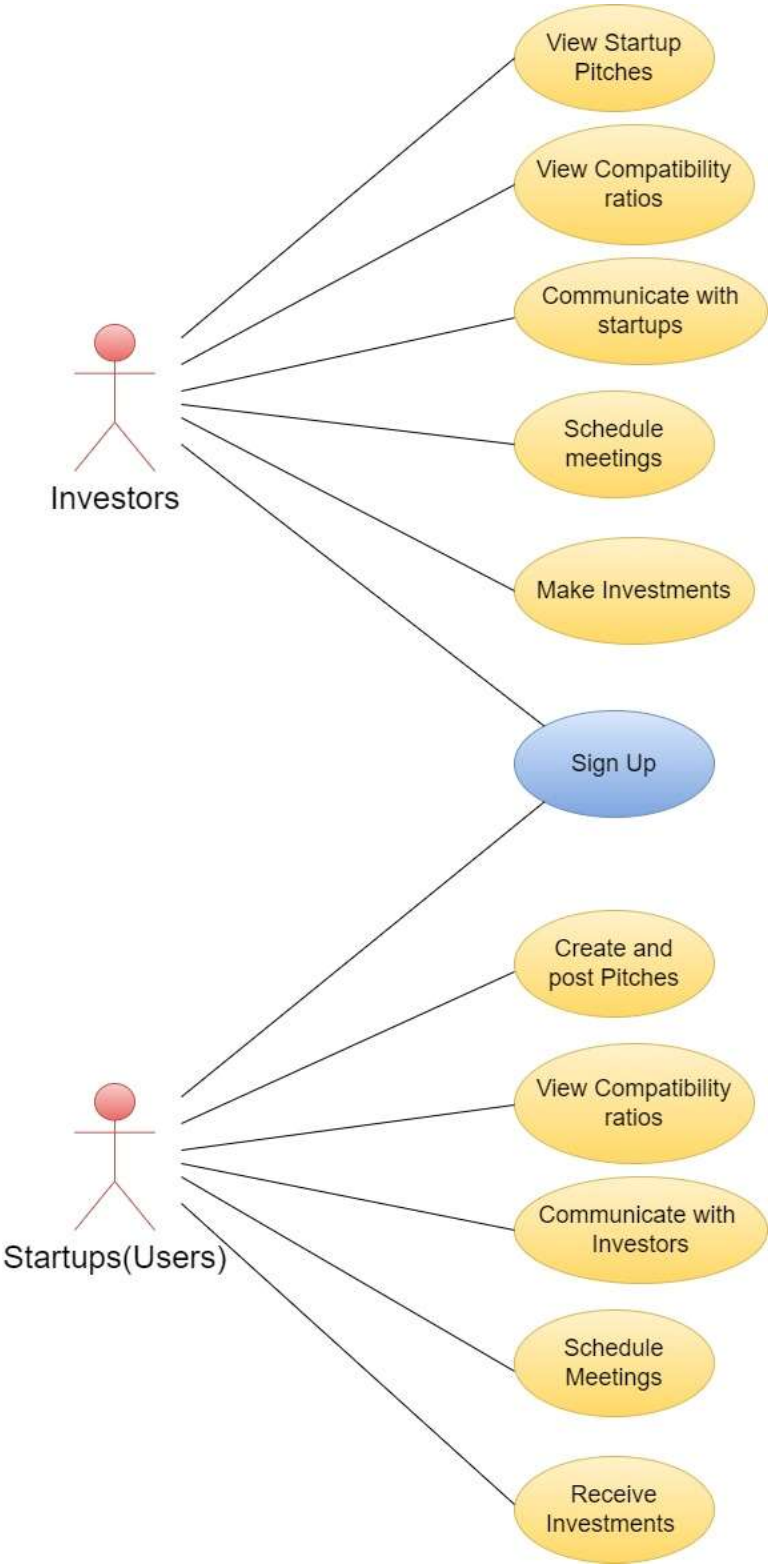


Fig 3.2 UML diagram

3.3 DATABASE DESIGN

3.1 Table Design

Table 3.1 chat_users

Sno.	Attributes	Data type	Description	Constraint
1	Notification_pushed	Varchar(25)	What notification is to be sent	Not null
2	Pushed_mail	Date & time	Date and time of sending the notification	Not null
3	Receiver_mail	Varchar(25)	Email of the receiver	Not null
4	Sender_mail	Varchar(25)	Email of the sender	PK
5	Sender_name	Varchar(25)	Name of the sender	Not null
6	Sender_pic	Image	Profile picture of the sender	null

Table 3.2 Company match

Sno.	Attributes	Data type	Description	Constraint
1	location	Char	Company's location	Not null
2	mail	Varchar(50)	Company email id	PK
3	sector	Varchar(50)	Company sector	Not null
4	size	Varchar(100)	Size of the investment required	Not null
5	stage	Varchar	Company growth stage	Not null

Table 3.3 investors

Sno.	Attributes	Data type	Description	Constraint
1	Annual Revenue	Varchar(50)	Revenue of the investor	Not null
2	bio	Varchar(100)	Brief description about the investor	Null
3	I_cpassword	Varchar(50)	Confirm password of the investor	Not null
4	I_display_name	Varchar(50)	Name of the investor	Not null
5	I_email	Varchar(50)	Investor's email id	PK
6	I_password	Varchar(50)	Password	Not null
7	I_phone_number	Integer	Investor phone number	Not null
8	I_pic	Image	Profile picture	Null
9	Interested_field	Varchar	Investor's interested field	Not null
10	Inv_sector	Varchar	Investor's interested sector	Not null
11	Inv_size	Varchar	Investment size preferred	Not null
12	Inv_stage	Varchar	Preferred company growth stage	Not null
13	Occupation	Varchar	Investor occupation	Not null
14	Previous_investment	Varchar	Previous investments	Not null

Table 3.4 my_requests_company

Sno.	Attributes	Data type	Description	Constraint
1	C_mail	Varchar(50)	Company's email	PK
2	I_mail	Varchar	Investors email	Not null
3	I_name	Varchar	Investor name	Not null
4	I_pic	image	Investor's profile pic	Not null
5	time	Date and time	Time and date the request was received	Not null

Table 3.5 my_requests_investor

Sno.	Attributes	Data type	Description	Constraint
1	C_mail	Varchar(50)	Company email	Not null
2	C_name	Varchar(50)	Name of the comapany	Not null
3	C_pic	image	Company profile picture	Not null
4	I_mail	Varchar(150)	Investor email	PK
5	time	Date time	Time and date the request was received	Not null

Table 3.6 post investor

Sno.	Attributes	Data type	Description	Constraint
1	Investor_email	Varchar(50)	Email id of the investor	PK
2	Post_description	Varchar(50)	Description added to the post	Not null

3	Post_image	image	Image posted	Not null
4	Post_inv_pic	image	Profile picture of investor	Not null
5	Post_investor_name	Varchar	Name of the investor	Not null
6	Post_occupation	Varchar	Occupation of the investor	Not null
7	Time_posted	Date and time	Date and Time of the post	Not null

Table 3.7 post user

Sno.	Attributes	Data type	Description	Constraint
1	Post_video	Video	Video posted	Not null
2	Time_posted	Date and time	Date and time of the post	Not null
3	Upost_description	Varchar	Description of the post	Not null
4	Upost_title	Varchar	Title of the post	Not null

Table 3.8 post user video

Sno.	Attributes	Data type	Description	Constraint
1	Com_email	Varchar(50)	Email id of the company	PK
2	Post_user	Varchar(50)	User id	Not null
3	Post_video	video	Video posted	Not null
4	Time_posted	Date and time	Time the post was made	Not null
5	U_video_description	Varchar	Description of the post	Not null

Table 3.9 request_inv

Sno.	Attributes	Data type	Description	Constraint
1	Notification_message	Varchar(50)	Notification message	Not null
2	Receiver_email	Varchar(50)	Email of receiver	Not null
3	Sender_mail	Varchar	Email of sender	PK
4	Sender_name	Varchar	Name of sender	Not null
5	Sender_pic	Image	Profile pic of sender	Not null
6	Time	Date and time	Date and time when notification was sent	Not null

Table 3.10 Users_1

Sno.	Attributes	Data type	Description	Constraint
1	annualRevenue	Varchar(50)	Annual revenue of company	Not null
2	C_location	Varchar(50)	Location of the company	Not null
3	C_sector	Varchar	Sector of the company	Not null
4	C_size	Varchar	Size of the company	Not null
5	C_stage	Varchar	Company growth stage	Not null
6	Customercount	Varchar	Number of customers in the company	Not null
7	Description	Varchar	About the company	Not null
8	Exit_strategy	Varchar	Exit strategy	Not null

9	Expected_count	Varchar	Expected customer count	Not null
10	Funduse	Varchar	What the investor money will be used for	Not null
11	Fundingrequirement	Varchar	How much funding is required	Not null
12	Marketsizeaim	Varchar	Market size aim	Not null
13	Marketsize	Varchar	Current market size	Not null
14	patentname	Varchar	Patent name	Not null
15	Patentnumber	Varchar	Patent number	Not null
16	Primaryrevenue	Varchar	Source of primary revenue	Not null
17	Proposedtimeline	Varchar	Proposed timeline	Not null
18	Target_customer	Varchar	Who are the target customers	Not null
29	Trademarknumber	Varchar	Trademark number	Not null
20	trademarkname	Varchar	Trademark name	Not null
21	U_cpassword	Varchar	User /company confirm password	Not null
22	U_displayname	Varchar	Company name	Not null
23	U_email	Varchar	Company email	PK
24	U_password	Varchar	Company password	Not null
25	U_phone number	Integer	Phone number	Not null
26	U_profile_pic	Image	Profile picture	Not null

3.5.2 Data Integrity and Constraints

Data integrity constraints within LightHouse are meticulously upheld to ensure the reliability and accuracy of the information stored and presented on the platform. Measures such as encryption of backend processes bolster security, safeguarding data from unauthorized access. Additionally, data undergoes rigorous standardization processes, converting raw information into a uniform format suitable for analysis. Through normalization principles, duplication of data within the database is mitigated, maintaining consistency and coherence. These constraints, coupled with compliance with regulatory standards and regular audits, fortify LightHouse's commitment to data integrity, fostering trust and confidence among users navigating the dynamic landscape of startup investment.

4. IMPLEMENTATION

This chapter consists of the coding standards used, the coding details, the code as well as the output of the desired implementation.

4.1 CODING STANDARDS

A coding standard gives a uniform appearance to the codes written by different engineers. It improves readability, and maintainability of the code and it reduces complexity also. It helps in code reuse and helps to detect errors easily. It promotes sound programming practices and increases efficiency of the programmers.

The following coding standards are followed:

- **Indenting** - Used an indent of 4 spaces and didn't use any tab because different computers use different setting for tab.
- **Variable Names** - Used all lowercase letters and used '_' as the word separator.
- **Control Structures -** These include if, for, while, switch, etc. Control statements have one space between the control keywords, to distinguish them from function calls.
- **Function Calls** - Functions have been called with no spaces between the function name, the opening parenthesis, and the first parameter; spaces between commas and each parameter, and no space between the last parameter, the closing parenthesis, and the semicolon.
- **Docstrings -** There are both single and multi-line docstrings that can be used in Python. However, the single line comment fits in one-line, triple quotes are used in both cases. These are used to define a particular program or define a particular function.
- **Comments -** There are also various types and conditions that are followed that can be of great help from programs and user's point of view. Comments should form complete sentences. If a comment is a full sentence, its first word should be capitalized, unless it is an identifier that begins with a lowercase letter. In short comments, the period at the end can be omitted. In block comments, there are more than one paragraphs and each sentence must end with a period. Block comments and inline comments can be written followed by a single '#'.
- **Alignment of Declaration Blocks** - Block of declarations are be aligned.

- **One Statement Per Line** - There is only one statement per line unless the statements are very closely related.

4.2 CODING DETAILS

Homepageuser :

```
import '/auth/firebase_auth/auth_util.dart';
import '/backend/backend.dart';
import '/flutter_flow/flutter_flow_animations.dart';
import '/flutter_flow/flutter_flow_expanded_image_view.dart';
import '/flutter_flow/flutter_flow_icon_button.dart';
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';
import '/flutter_flow/flutter_flow_widgets.dart';
import '/investor/side_nav04_copy/side_nav04_copy_widget.dart';
import '/pages/navbar/ihome/ihome_widget.dart';
import 'package:flutter/material.dart';
import 'package:flutter/scheduler.dart';
import 'package:flutter_animate/flutter_animate.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:page_transition/page_transition.dart';
import 'package:provider/provider.dart';

import 'homepageinvestor_model.dart';
export 'homepageinvestor_model.dart';

class HomepageinvestorWidget extends StatefulWidget {
  const HomepageinvestorWidget({super.key});

  @override
  State<HomepageinvestorWidget> createState() =>
_HomepageinvestorWidgetState();
}
```

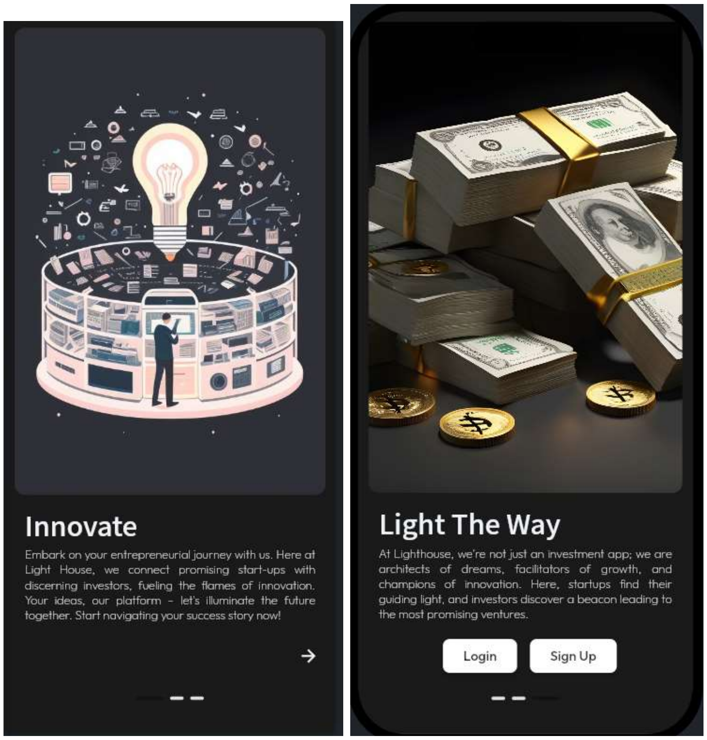
```
class _HomepageinvestorWidgetState extends State<HomepageinvestorWidget>
    with TickerProviderStateMixin {
  late HomepageinvestorModel _model;

  final scaffoldKey = GlobalKey<ScaffoldState>();

  final animationsMap = <String, AnimationInfo>{};

  @override
  void initState() {
    super.initState();
    _model = createModel(context, () => HomepageinvestorModel());

    animationsMap.addAll({
      'textOnPageLoadAnimation1': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          MoveEffect(
            curve: Curves.easeIn,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(0, 15),
            end: Offset(0, 0),
          ),
        ],
      ),
      'textOnPageLoadAnimation2': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(30, 0),
            end: Offset(0, 0),
          ),
        ],
      ),
      'textOnPageLoadAnimation3': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(40, 0),
            end: Offset(0, 0),
          ),
        ],
      ),
      'containerOnPageLoadAnimation1': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(60, 0),
            end: Offset(0, 0),
          ),
        ],
      ),
      'containerOnPageLoadAnimation2': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(60, 0),
            end: Offset(0, 0),
          ),
        ],
      ),
      'textOnPageLoadAnimation4': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(0, 40),
            end: Offset(0, 0),
          ),
        ],
      ),
      'textOnPageLoadAnimation5': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(0, 60),
            end: Offset(0, 0),
          ),
        ],
      ),
      'listViewOnPageLoadAnimation': AnimationInfo(
        trigger: AnimationTrigger.onPageLoad,
        effectsBuilder: () => [
          FadeEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: 0,
            end: 1,
          ),
          MoveEffect(
            curve: Curves.easeInOut,
            delay: 0.0.ms,
            duration: 600.0.ms,
            begin: Offset(0, 80),
            end: Offset(0, 0),
          ),
        ],
      ),
    });

    WidgetsBinding.instance.addPostFrameCallback((_) => setState(() {}));
  }

  @override
  void dispose() {
    _model.dispose();

    super.dispose();
  }
```

```
@override
Widget build(BuildContext context) {
 return GestureDetector(
  onTap: () => _model.unfocusNode.canRequestFocus
    ? FocusScope.of(context).requestFocus(_model.unfocusNode)
    : FocusScope.of(context).unfocus(),
  child: Scaffold(
   key: scaffoldKey,
   backgroundColor: Color(0xFFF1F4F8),
   body: Stack(
    children: [
     Stack(
      children: [
       Stack(
        children: [
         Stack(
          children: [
           StreamBuilder<List<InvestorsRecord>>(
            stream: queryInvestorsRecord(
             queryBuilder: (investorsRecord) =>
               investorsRecord.where(
              'i_email',
              isEqualTo: currentUserEmail,
             ),
             singleRecord: true,
            ),
            builder: (context, snapshot) {
             // Customize what your widget looks like when it's loading.
             if (!snapshot.hasData) {
              return Center(
               child: SizedBox(
                width: 50,
                height: 50,
      child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
          Color(0xFFF1F4F8),
        ),
      ),
    ),
  );
}
List<InvestorsRecord> columnInvestorsRecordList =
    snapshot.data!;
final columnInvestorsRecord =
    columnInvestorsRecordList.isNotEmpty
        ? columnInvestorsRecordList.first
        : null;
return Column(
  mainAxisSize: MainAxisSize.max,
  children: [
    Expanded(
      child: Container(
        height: 500,
        child: Stack(
          alignment: AlignmentDirectional(0, -1),
          children: [
            Align(
              alignment:
                  AlignmentDirectional(0.05, -1),
              child: Image.asset(
                'assets/images/timw.jpeg',
                width: double.infinity,
                height: 500,
                fit: BoxFit.cover,
              ),
            ),
            Container(
              width: double.infinity,
              height: 500,
              decoration: BoxDecoration(
                color: Color(0x8D090F13),
              ),
            ),
            Align(
              alignment: AlignmentDirectional(0, 0),
              child: SingleChildScrollView(
                child: Column(
                  mainAxisSize: MainAxisSize.max,
                  crossAxisAlignment:
                      CrossAxisAlignment.start,
                  children: [
                    Padding(
                      padding: EdgeInsetsDirectional
                          .fromSTEB(340, 20, 0, 0),
                      child: FlutterFlowIconButton(
                        borderRadius: 20,
                        borderWidth: 1,
                        buttonSize: 40,
                        icon: Icon(
                          Icons.search_outlined,
                          color:
                              FlutterFlowTheme.of(
                                      context)
                                  .alternate,
                          size: 30,
                        ),
                        onPressed: () async {
                          context.pushNamed(
                              'search_investor');
                        },
                      ),
                    ),
                    Padding(
                      padding: EdgeInsetsDirectional
                          .fromSTEB(16, 24, 16, 44),
                      child: Text(
                        'Discover Tomorrow\'s Success Today.',
                        style: FlutterFlowTheme.of(
                                context)
                            .displaySmall
                            .override(
                              fontFamily: 'Outfit',
                              color: Colors.white,
                              fontSize: 36,
                              letterSpacing: 0,
                              fontWeight:
                                  FontWeight.w600,
                            ),
                      ).animateOnPageLoad(animationsMap[
                          'textOnPageLoadAnimation1']!),
                    ),
                    Padding(
                      padding: EdgeInsetsDirectional
                          .fromSTEB(0, 32, 0, 0),
                      child: Container(
                        width: double.infinity,
                        height: 611,
                        decoration: BoxDecoration(
                          color: Colors.white,
                          borderRadius:
                              BorderRadius.only(
                            bottomLeft:
                                Radius.circular(0),
                            bottomRight:
                                Radius.circular(0),
                            topLeft:
                                Radius.circular(16),
                            topRight:
                                Radius.circular(16),
                          ),
                        ),
                        child: Stack(
                          children: [
                            Padding(
                              padding:
                                  EdgeInsetsDirectional
                                      .fromSTEB(0,
                                          8, 0, 24),
                              child:
                                  SingleChildScrollView(
                                child: Column(
                                  mainAxisSize:
                                      MainAxisSize
                                          .max,
                                  crossAxisAlignment:
                                      CrossAxisAlignment
                                          .start,
                                  children: [
                                    Padding(
                                      padding: EdgeInsetsDirectional
                                          .fromSTEB(
                                              16,
                                              16,
                                              16,
                                              0),
                                      child: Text(
                                        'Start Investing ',
                                        style: FlutterFlowTheme.of(
                                                context)
                                .headlineMedium
                                .override(
                                  fontFamily:
                                      'Outfit',
                                  color:
                                      Color(0xFF14181B),
                                  fontSize:
                                      24,
                                  letterSpacing:
                                      0,
                                  fontWeight:
                                      FontWeight.normal,
                                ),
                          ).animateOnPageLoad(
                              animationsMap[
                                  'textOnPageLoadAnimation2']!),
                        ),
                        Padding(
                          padding: EdgeInsetsDirectional
                              .fromSTEB(
                                  16,
                                  4,
                                  16,
                                  0),
                          child: Text(
                            'Top compatible ',
                            style: FlutterFlowTheme.of(
                                    context)
                                .labelMedium
                                .override(
                                  fontFamily:
                                      'Plus Jakarta Sans',
                                  color:
                                      Color(0xFF57636C),
                                  fontSize:
                                      14,
                                  letterSpacing:
                                      0,
                                  fontWeight:
                                      FontWeight.normal,
                                ),
                          ).animateOnPageLoad(
                              animationsMap[
                                  'textOnPageLoadAnimation3']!),
                        ),
                        Padding(
                          padding: EdgeInsetsDirectional
                              .fromSTEB(
                                  0,
                                  12,
                                  0,
                                  0),
                          child:
                              Container(
                            width: double
                                .infinity,
                            height: 210,
                            decoration:
                                BoxDecoration(
                              color: Colors
                                  .white,
                            ),
                            child: StreamBuilder<
                                List<
                                    Users1Record>>(
                              stream:
                                  queryUsers1Record(),
                              builder:
```

```
  (context,
      snapshot) {
// Customize what your widget looks like
when it's loading.
if (!snapshot
    .hasData) {
  return Center(
    child:
        SizedBox(
      width: 50,
      height: 50,
      child: CircularProgressIndicator(
        valueColor:
AlwaysStoppedAnimation<Color>(
          Color(0xFFF1F4F8),
        ),
      ),
    ),
  );
}
List<Users1Record>
    listViewUsers1RecordList =
    snapshot.data!;
return ListView
    .builder(
  padding:
      EdgeInsets.zero,
  primary:
      false,
  shrinkWrap:
      true,
  scrollDirection:
      Axis.horizontal,
  itemCount:
```

```
                                            listViewUsers1RecordList.length,
                                        itemBuilder:
                                            (context, listViewIndex) {
                                          final listViewUsers1Record =

listViewUsers1RecordList[listViewIndex];
                                          return Visibility(
                                            visible:
(columnInvestorsRecord?.invSector == listViewUsers1Record.cSector) ||
(columnInvestorsRecord?.invStage == listViewUsers1Record.cStage) ||
(columnInvestorsRecord?.invSize == listViewUsers1Record.cSize) ||
(columnInvestorsRecord?.invLocation == listViewUsers1Record.cLocation),
                                            child: Padding(
                                              padding:
EdgeInsetsDirectional.fromSTEB(16, 8, 0, 12),
                                              child: InkWell(
                                                splashColor: Colors.transparent,
                                                focusColor: Colors.transparent,
                                                hoverColor: Colors.transparent,
                                                highlightColor: Colors.transparent,
                                                onTap: () async {
                                                  context.pushNamed(
                                                    'user_page_invpointov',
                                                    queryParameters: {
                                                      'userMail': serializeParam(

listViewUsers1Record.uEmail,
                                                        ParamType.String,
                                                      ),
                                                      'inMail': serializeParam(

columnInvestorsRecord?.iEmail,
                                                        ParamType.String,
                                                      ),
                                                      'invPIc': serializeParam(
                                                        columnInvestorsRecord?.iPic,
                                                        ParamType.String,
                                                      ),
                                                      'invName': serializeParam(

columnInvestorsRecord?.iDisplayName,
                                                        ParamType.String,
                                                      ),
                                                    }.withoutNulls,
                                                  );
                                                },
                                                child: Container(
                                                  width: 270,
                                                  height: 100,
                                                  decoration: BoxDecoration(
                                                    color: Colors.white,
                                                    boxShadow: [
                                                      BoxShadow(
                                                        blurRadius: 8,
                                                        color: Color(0x230F1113),
                                                        offset: Offset(
                                                          0.0,
                                                          4,
                                                        ),
                                                      )
                                                    ],
                                                    borderRadius:
BorderRadius.circular(12),
                                                    border: Border.all(
                                                      color: Color(0xFFF1F4F8),
                                                      width: 1,
                                                    ),
                                                  ),
                                                child: Column(
                                                  mainAxisSize:
MainAxisSize.max,
                                                  children: [
                                                    Expanded(
                                                      child: ClipRRect(
                                                        borderRadius:
BorderRadius.only(
                                                          bottomLeft:
Radius.circular(0),
                                                          bottomRight:
Radius.circular(0),
                                                          topLeft:
Radius.circular(12),
                                                          topRight:
Radius.circular(12),
                                                        ),
                                                        child: Image.network(

listViewUsers1Record.uProfilePic,
                                                          width: double.infinity,
                                                          height: 124,
                                                          fit: BoxFit.contain,
                                                        ),
                                                      ),
                                                    ),
                                                    Padding(
                                                      padding:
EdgeInsetsDirectional.fromSTEB(16, 12, 16, 12),
                                                      child: Row(
                                                        mainAxisSize:
MainAxisSize.max,
                                                        mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                                              children: [
                                               Column(
                                                mainAxisSize:
MainAxisSize.max,
                                                crossAxisAlignment:
CrossAxisAlignment.start,
                                                children: [
                                                 Text(

listViewUsers1Record.uDisplayName,
                                                  style:
FlutterFlowTheme.of(context).bodyLarge.override(
                                                     fontFamily: 'Plus
Jakarta Sans',
                                                     color:
Color(0xFF14181B),
                                                     fontSize: 16,
                                                     letterSpacing: 0,
                                                     fontWeight:
FontWeight.normal,
                                                    ),
                                                 ),
                                                 Padding(
                                                  padding:
EdgeInsetsDirectional.fromSTEB(0, 8, 0, 0),
                                                  child: Row(
                                                   mainAxisSize:
MainAxisSize.max,
                                                   children: [
                                                    Text(

listViewUsers1Record.cSector,
                                                     style:
FlutterFlowTheme.of(context).bodyMedium.override(
                                                        fontFamily:
'Manrope',
                                                        letterSpacing: 0,
                                                       ),
                                                    ),
                                                   ],
                                                  ),
                                                 ),
                                                ],
                                               ),
                                               Expanded(
                                                child: Padding(
                                                 padding:
EdgeInsetsDirectional.fromSTEB(10, 0, 0, 0),
                                                 child: Text(

valueOrDefault<String>(
                                                   () {
                                                    if
((columnInvestorsRecord?.invSector == listViewUsers1Record.cSector) &&
(columnInvestorsRecord?.invStage == listViewUsers1Record.cStage) &&
(columnInvestorsRecord?.invSize == listViewUsers1Record.cSize) &&
(columnInvestorsRecord?.invLocation == listViewUsers1Record.cLocation)) {
                                                     return '99%';
                                                    } else if
(((columnInvestorsRecord?.invSector == listViewUsers1Record.cSector) &&
(columnInvestorsRecord?.invStage == listViewUsers1Record.cStage) &&
(columnInvestorsRecord?.invSize == listViewUsers1Record.cSize)) ||
((columnInvestorsRecord?.invStage == listViewUsers1Record.cStage) &&
(columnInvestorsRecord?.invSize == listViewUsers1Record.cSize) &&
(columnInvestorsRecord?.invLocation == listViewUsers1Record.cLocation)) ||
((columnInvestorsRecord?.invSector == listViewUsers1Record.cSector) &&
(columnInvestorsRecord?.invSize == listViewUsers1Record.cSize) &&
(columnInvestorsRecord?.invLocation == listViewUsers1Record.cLocation)) || ((
),
Align(
  alignment: AlignmentDirectional(
      -0.91, -0.99),
  child: Padding(
    padding:
        EdgeInsetsDirectional.fromSTEB(
            0, 20, 0, 0),
    child: FlutterFlowIconButton(
      borderColor: Color(0x009489F5),
      borderRadius: 20,
      borderWidth: 1,
      buttonSize: 40,
      fillColor: Color(0x009489F5),
      icon: Icon(
        Icons.toc_outlined,
        color:
            FlutterFlowTheme.of(context)
                .info,
        size: 30,
      ),
      onPressed: () async {
        await showModalBottomSheet(
          isScrollControlled: true,
          backgroundColor:
              Colors.transparent,
          enableDrag: false,
          context: context,
          builder: (context) {
            return GestureDetector(
              onTap: () => _model
                      .unfocusNode
                      .canRequestFocus
                  ? FocusScope.of(
                          context)
                        .requestFocus(_model
                          .unfocusNode)
                      : FocusScope.of(
                          context)
                        .unfocus(),
                    child: Padding(
                     padding: MediaQuery
                       .viewInsetsOf(
                         context),
                     child:
                       SideNav04CopyWidget(),
                    ),
                   );
                  },
                 ).then((value) =>
                   safeSetState(() {}));
                },
               ),
              ),
             ),
            ],
           ),
          ),
         ),
        ],
       );
      },
     ),
    ],
   ),
  ],
 ),
],
            ),
            Align(
              alignment: AlignmentDirectional(-0.99, 1),
              child: Container(
                width: 398,
                height: 68,
                decoration: BoxDecoration(
                  color: FlutterFlowTheme.of(context).info,
                ),
                child: wrapWithModel(
                  model: _model.ihomeModel,
                  updateCallback: () => setState(() {}),
                  child: IhomeWidget(),
                ),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

4.3 SCREENSHOTS



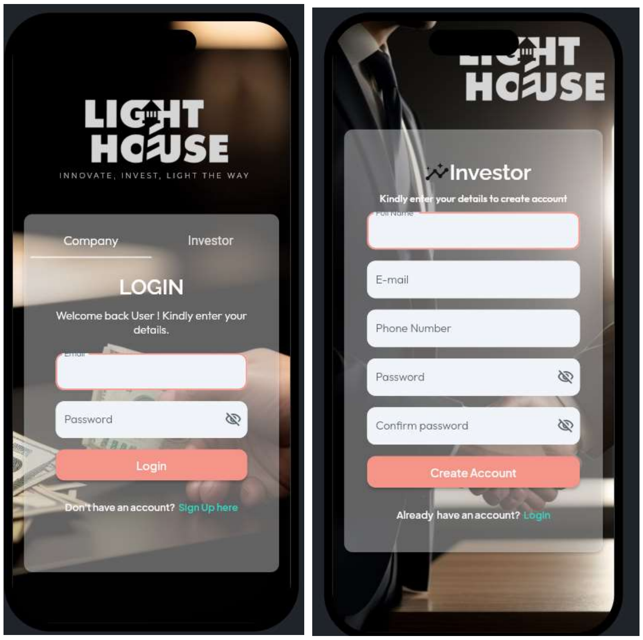
Innovate

Embark on your entrepreneurial journey with us. Here at Light House, we connect promising start-ups with discerning investors, fueling the flames of innovation. Your ideas, our platform – let's illuminate the future together. Start navigating your success story now!

Light The Way

At Lighthouse, we're not just an investment app; we are architects of dreams, facilitators of growth, and champions of innovation. Here, startups find their guiding light, and investors discover a beacon leading to the most promising ventures.

Fig 4.1 Initial page



LIGHT HOUSE

INNOVATE, INVEST, LIGHT THE WAY

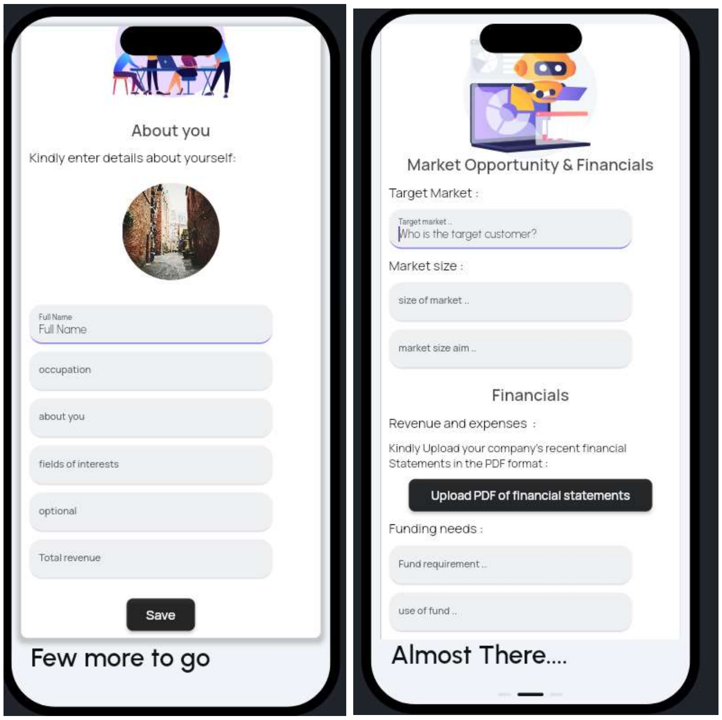
LOGIN

Welcome back User ! Kindly enter your details.

Investor

Kindly enter your details to create account

Fig 4.2 login/sign up



About you

Kindly enter details about yourself:

Full Name
Full Name

occupation

about you

fields of interests

optional

Total revenue

Save

Few more to go

Market Opportunity & Financials

Target Market :

Target market ..
Who is the target customer?

Market size :

size of market ..

market size aim ..

Financials

Revenue and expenses :

Kindly Upload your company's recent financial Statements in the PDF format :

Upload PDF of financial statements

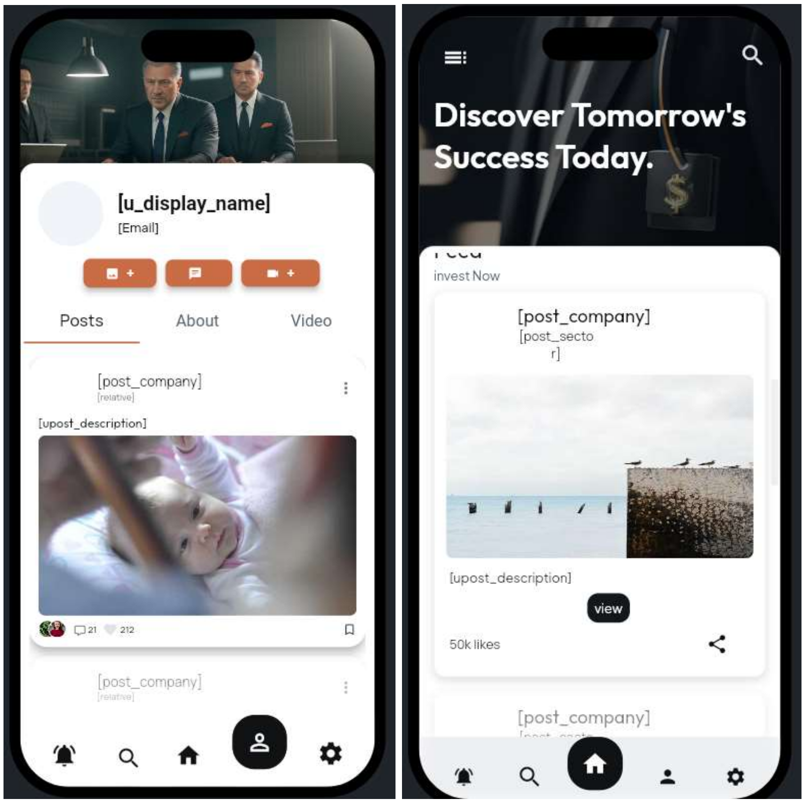
Funding needs :

Fund requirement ..

use of fund ..

Almost There....

Fig 4.3 details page



[u_display_name]
[Email]

Posts About Video

[post_company]
[relative]

[upost_description]

21 212

[post_company]
[relative]

Discover Tomorrow's Success Today.

Feed
invest Now

[post_company]
[post_secto
r]

[upost_description]

view

50k likes

[post_company]

Fig 4.4 Home page

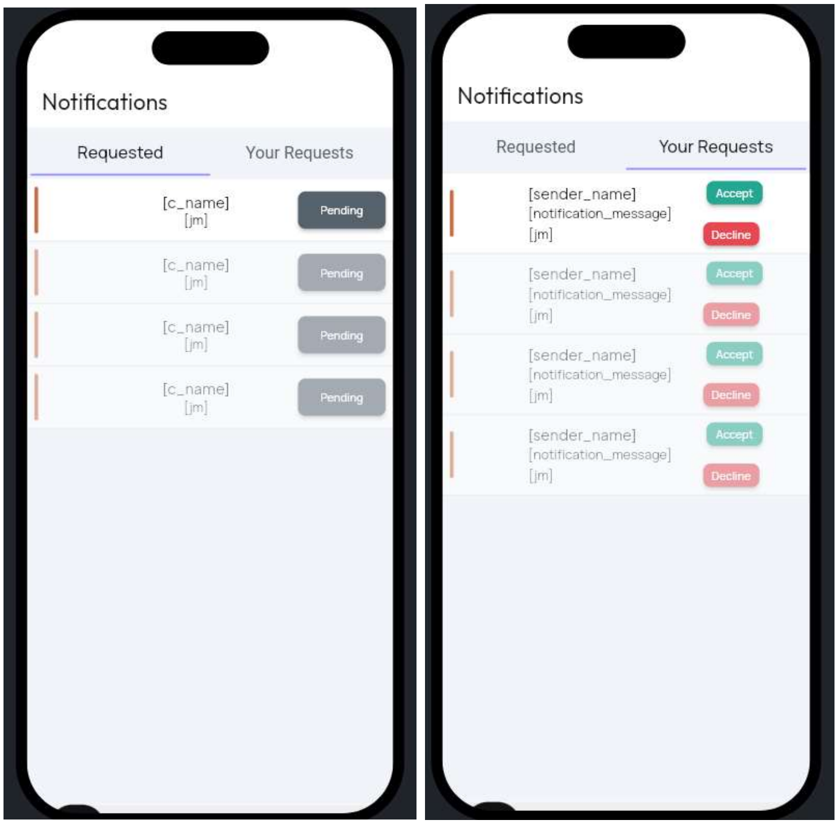


Fig 4.5 notifications

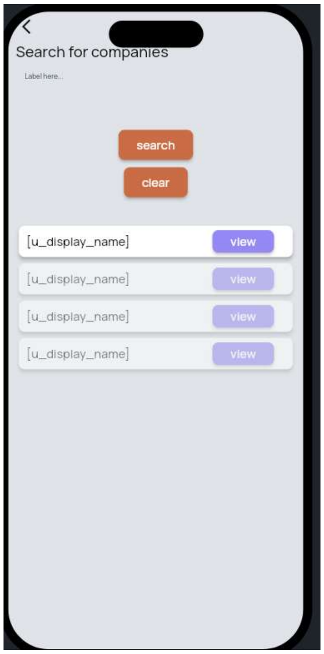


Fig 4.6 search

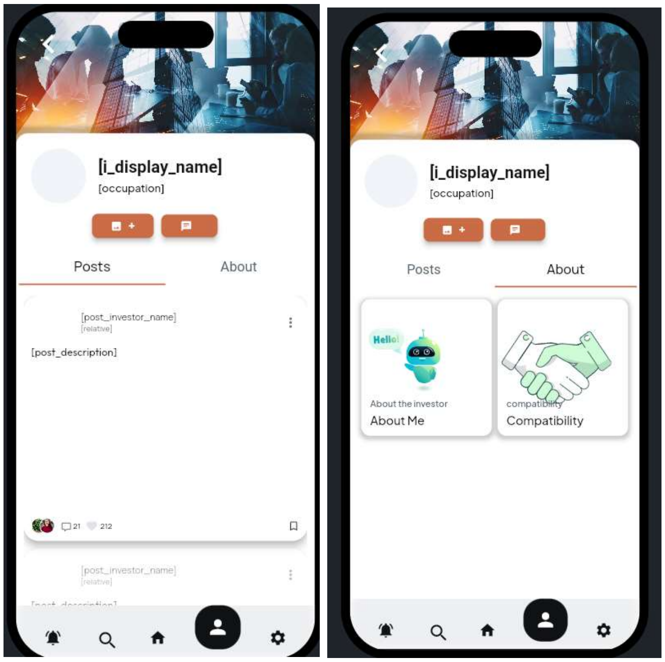
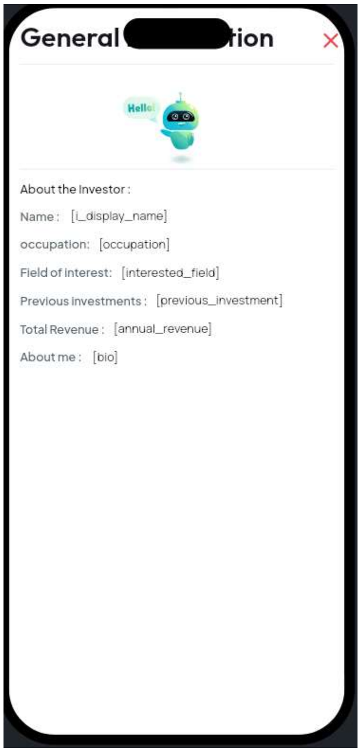


Fig 4.7 profile page



About the Investor :

Name : [i_display_name]

occupation: [occupation]

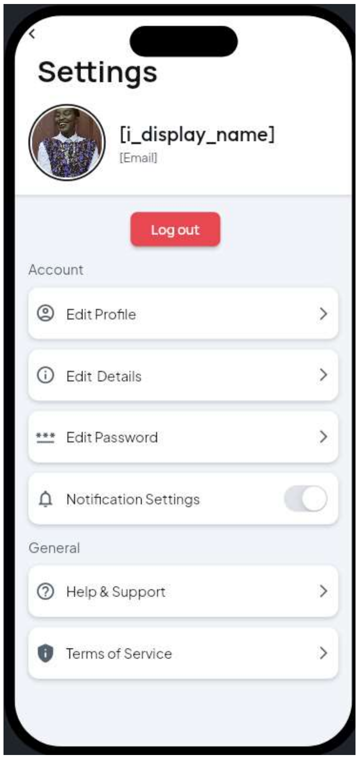
Field of Interest: [interested_field]

Previous Investments : [previous_investment]

Total Revenue : [annual_revenue]

About me : [bio]

Fig 4.8 information page



Settings

[i_display_name]
[Email]

Log out

Account

Edit Profile

Edit Details

Edit Password

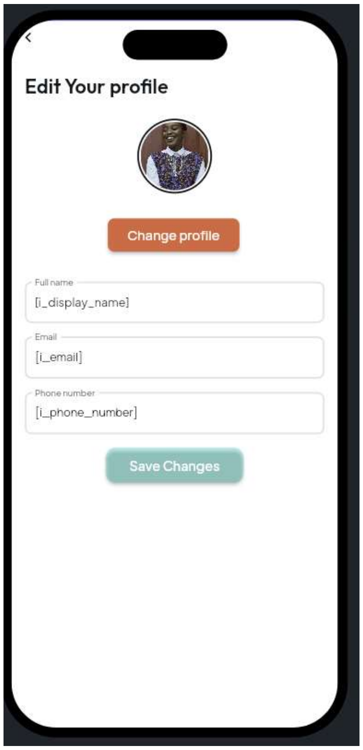
Notification Settings

General

Help & Support

Terms of Service

Fig 4.9 settings



Edit Your profile

Change profile

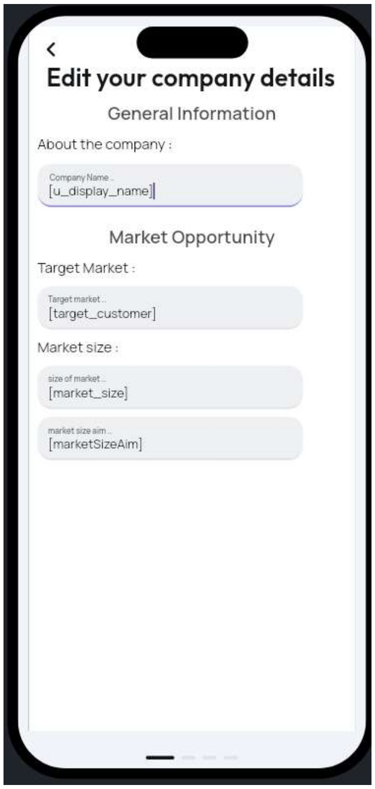
Full name
[i_display_name]

Email
[i_email]

Phone number
[i_phone_number]

Save Changes

Fig 4.10 Edit profile



Edit your company details

General Information

About the company :

Company Name ..
[u_display_name]

Market Opportunity

Target Market :

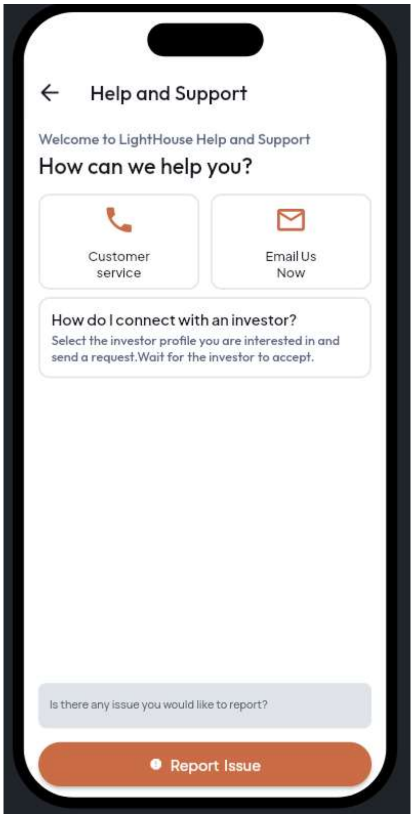
Target market ..
[target_customer]

Market size :

size of market ..
[market_size]

market size aim ..
[marketSizeAim]

Fig 4.11 edit company details



Welcome to LightHouse Help and Support

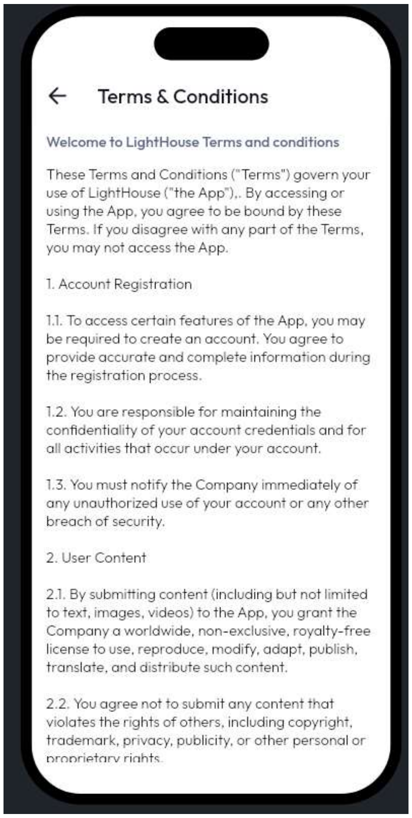
How can we help you?

How do I connect with an investor?

Select the investor profile you are interested in and send a request.Wait for the investor to accept.

Is there any issue you would like to report?

Fig 4.12 help and support



Welcome to LightHouse Terms and conditions

These Terms and Conditions ("Terms") govern your use of LightHouse ("the App"),. By accessing or using the App, you agree to be bound by these Terms. If you disagree with any part of the Terms, you may not access the App.

1. Account Registration

1.1. To access certain features of the App, you may be required to create an account. You agree to provide accurate and complete information during the registration process.

1.2. You are responsible for maintaining the confidentiality of your account credentials and for all activities that occur under your account.

1.3. You must notify the Company immediately of any unauthorized use of your account or any other breach of security.

2. User Content

2.1. By submitting content (including but not limited to text, images, videos) to the App, you grant the Company a worldwide, non-exclusive, royalty-free license to use, reproduce, modify, adapt, publish, translate, and distribute such content.

2.2. You agree not to submit any content that violates the rights of others, including copyright, trademark, privacy, publicity, or other personal or proprietary rights.

Fig 4.13 terms and conditions

5. TESTING

5.1 TEST STRATEGIES

5.1.1 System Testing

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. Test case design focuses on set of techniques for the creation of test because that meet overall testing objectives. When a system is developed it is hoped that it performs properly. The main purpose of testing an information system Is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is a comprehensive evaluation of the programs, manual procedures, computer operations, and controls.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted by the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done systematically.

The process of analysis of the software item to detect the differences between existing or required conditions and evaluate the features of the software items. The thorough testing of the system before the release of the software becomes devoid of bugs and uses minimum space requirements as well as minimum time to perform. The test cases were selected beforehand with expected results recorded for comparison. The selection of the test cases is done vide the “White Box Testing” technique to check software requirement fulfillment to find a maximum number of errors with minimum effort and time. Although test cases are a designed by considering the cyclomatic complexity, and conditional test, still the software code is not in its optional form, as all other possible alternative parts in the software are not considered. At the integration level, the software will be passing to the third-party tests which would further enhance the software optimality and efficiency.

5.1.2 Test Data Implementation

The quality and standardization of the software /application package depends truly on the various predefined testing norms and on the performance of the software over those norms. There are various standards existing in the software industry the engineered end product

Strives to achieve viz. ISO 9002 SEI CMM Level5 etc. These standards are achieved only when the concerned software fulfils the tests as per the respective norms predefined in them vide the various test cases and parameters using the CASE topologies. Generally, Software is tested both on a stand-alone mode as well after integrating all the modules in the system vide deferent available testing methods/norms. The following Flow Graph methodology was used while testing the software:

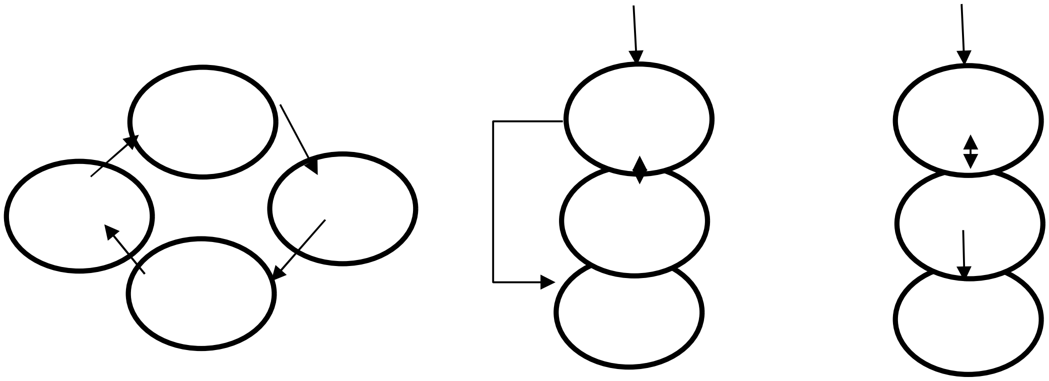


Fig 5.1 Flow graph

Here each circle represents one or more nonbranching procedural language or source code statements in Flow Graph. While performing Condition Testing Domain Testing methodology was selected. While performing Loop testing simple loops, concatenated loops, nested, and unstructured loops were tested thoroughly.

5.1.3 Test Characteristics

1. A good test has a high probability of finding an error.
2. A good test is not redundant.
3. A good test should be “best of breed “.
4. A good test should be neither too simple nor too complex.

5.1.4 Black box Testing

The method of Black Box Testing is used by the software engineer to derive the required results of the test cases:

1. Black Box Testing alludes to test that are conducted at the software interface.
2. A Black Box Test examines some fundamental aspect of a system with little regard for the internal logic structure of the software.
3. A limited number of important logical paths can be selected and exercised.
4. Important data structure can be probed for validity.

Black box testing was performed to find errors in the following categories:

- Incorrect or missing functions
- Graphics errors.
- Errors in data in string format.
- Error in data in integer format.
- File error.
- Memory access error
- Variable error.
- Performance error.

5.1.5 White box Testing

White Box Testing is sometimes called as Glass Box Testing. Using White Box Testing methods, the software engineer can derive the following test cases:

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds.
4. Exercise internal data structures to ensure validity.

In white Box Testing efforts were made to handle the following:

- Number of input parameters equal to number of arguments.
- Parameters and arguments attributes match.
- The number of arguments transmitted is called modules equal to attributes of the parameter.

- Unit system of argument transmitted is called modules equal unit system of parameter.
- Number of attributes and order of arguments to build in functions correct.
- Any references to parameters not associated to build in functions correct.
- Input only arguments altered.
- Global variable definition consistent across module.
- Files attributes correct.
- Format specification matches I/O specification.
- Files opened before use.
- File closed while working.
- I/O errors handled.
- Any textual errors in output information.

Unit Testing

The unit testing is performed to test the validity of the individual units. This is done in the coding phase with the interactive testing. Thus, it itself constitutes a majority of functionality test for each logical unit.

Integrity Testing

When all the development of all the units or modules is completed and integrated, the integrity test phase is started. In this phase, the interface between the modules are tested. This phase basically verifies whether inter module exchange of information and events are as per required system behavior.

Validation Testing

Tests were performed to find conformity with the requirements. Plans and procedures were designed to ensure that all the functional requirements are satisfied. The software was alpha-tested. There are two goals in preparing test plans. Firstly, a properly detailed test plan demonstrates that the program specifications are understood completely. Secondly, the test plan is used during program testing to prove the correctness of the program.

5.2 TEST CASES

Table 5.2.1 Test Cases

No	Module Name	Test Case No	Test Case Description	Expected Result
1	Startup Pitch Posting	TC1	Startups create and post pitches with images, video pitches, and graphs.	Pitch is successfully created and posted.
		TC2	Startups add new startup ideas to their pitches.	New startup idea is added to the pitch.
2	Compatibility Ratio Algorithm	TC3	Compatibility ratio is calculated for each pitch.	Each pitch displays a compatibility ratio.
		TC4	Investors view the compatibility ratio for a specific pitch.	Investors see the alignment measure clearly.
3	Tiered Investment Model	TC5	Accredited investors access exclusive benefits.	Accredited investors enjoy exclusive perks.
		TC6	Different tiers provide varying degrees of access and perks.	Each tier's benefits are clearly defined.
4	Industry-Specific Showcases	TC7	Industry-specific showcases and themes are introduced.	Startups across various sectors are visible.
		TC8	Startups from non-tech sectors gain equitable visibility.	Non-tech startups are equally showcased.
5	Direct Investment Feature	TC9	Investors make direct investments within the platform.	Investment process is streamlined.

		TC10	Startups receive direct investments through the platform.	Investment transactions are facilitated.
6	Communication Hub	TC11	Startups and investors engage through text, email, and scheduled meetings.	Communication options are accessible.
		TC12	Scheduled meetings are successfully arranged between startups and investors.	Collaboration between parties is smooth.
7	Global Expansion Strategies	TC13	Strategies for global expansion are implemented.	Platform is accessible to diverse regions.
		TC14	Regulatory challenges in different markets are addressed effectively.	Platform complies with diverse regulations.

5.3 TEST REPORTS

Table 5.3.1 Test Reports

Sl. No.	Test Case No.	Test Status	Test Result
1	TC1	Passed	Pitch creation successful
	TC2	Passed	New startup idea added
2	TC3	Passed	Compatibility ratio calculated
	TC4	Passed	Investors view ratio clearly
3	TC5	Passed	Accredited investors access perks
	TC6	Passed	Tier benefits clearly defined
4	TC7	Passed	Industry showcases introduced
	TC8	Passed	Non-tech startups visible

5	TC9	Passed	Direct investments made
	TC10	Passed	Startups receive investments
6	TC11	Passed	Communication options accessible
	TC12	Passed	Scheduled meetings arranged
7	TC13	Passed	Global expansion strategies implemented
	TC14	Passed	Regulatory challenges addressed

6. CONCLUSION

The major implementation and design issues along with advantages and disadvantages of the project is properly mentioned. The future scope of the project is also mentioned in this chapter.

6.1 DESIGN AND IMPLEMENTATION ISSUES

In the development journey of LightHouse, design and implementation challenges emerged, highlighting the critical role of robust design in the success of the platform.

6.1.1 Design Issues

During the design phase, several hurdles were encountered. Integrating the Compatibility Ratio Algorithm posed difficulties due to its innovative nature and the need for precise alignment with investor preferences. Initially, aligning features such as the Communication Hub for effortless engagement between startups and investors proved to be a design challenge. Additionally, devising Global Expansion Strategies demanded strategic planning to address regulatory challenges and tailor the platform for diverse markets. Despite these design complexities, LightHouse persevered, leveraging innovation and user-centric design principles to navigate through implementation challenges and emerge as a transformative force in the realm of startup investment.

6.1.2 Implementation Issues

Implementation of LightHouse faced challenges primarily in integrating the Compatibility Ratio Algorithm to ensure accurate alignment between startup pitches and investor preferences. Additionally, refining the Communication Hub for seamless interaction between stakeholders posed difficulties. Overcoming these hurdles required meticulous planning and adaptation of FlutterFlow and Firebase to meet the platform's unique requirements. Despite the challenges, LightHouse successfully executed its vision, leveraging innovative solutions and user-centric design principles to create a transformative platform for startup investment.

6.2 ADVANTAGES AND LIMITATIONS

6.2.1 Advantages

- Efficient pitch evaluation with the Compatibility Ratio Algorithm.
- Diverse investment opportunities through the Tiered Investment Model.
- Industry-specific visibility for startups across various sectors.
- Streamlined investment process with the Direct Investment Feature.
- Effective communication and collaboration via the Communication Hub.
-

6.2.2 Limitations

- Limited accessibility for users without internet connectivity.
- Potential exclusion of startups and investors not aligned with the platform's ecosystem.
- Compliance challenges with diverse regulatory frameworks across regions..

6.3 FUTURE AND SCOPE OF THE PROJECT

The future of LightHouse holds immense potential as it continues to evolve in the dynamic landscape of startup investment. With ongoing advancements in technology and an increasing emphasis on innovation, the platform is poised to expand its reach and impact. Future developments may include further enhancements to the Compatibility Ratio Algorithm, leveraging machine learning and data analytics to provide even more accurate insights for investors. Additionally, LightHouse can explore partnerships and collaborations to broaden its industry-specific showcases, catering to a wider array of startup sectors and fostering greater diversity in investment opportunities. As the entrepreneurial ecosystem continues to grow and evolve, LightHouse stands at the forefront, facilitating connections between visionary startups and discerning investors, driving innovation and growth in the global startup community.

Furthermore, the scope of LightHouse extends beyond its current functionalities, presenting opportunities for expansion and innovation. Potential avenues for growth include the integration of emerging technologies such as blockchain and decentralized finance (DeFi) to enhance transparency and security in investment transactions. Moreover, LightHouse can explore avenues for global expansion, tapping into untapped markets and catering to startups and investors from diverse regions. Additionally, the platform can explore the development of educational resources and mentorship programs to support startup founders in navigating the entrepreneurial journey. With a commitment to innovation and user-centric design, LightHouse is well-positioned to continue shaping the future of startup investment, empowering entrepreneurs to bring their visionary ideas to life and fueling economic growth and prosperity worldwide.

REFERENCES

1. Smith, John. (2023). "Revolutionizing Startup-Investor Connections: " *Journal of Entrepreneurial Finance*, vol. 15, no. 2, pp. 45-62.
2. Johnson, Emily. (2022). "Innovative Online Investment Platforms: A Comparative Analysis." *International Journal of Financial Innovation*, vol. 7, no. 1, pp. 103-120.
3. Williams, Michael. (2024). "The Role of Technology in Facilitating Startup-Investor Relationships." *Journal of Business and Technology*, vol. 10, no. 3, pp. 75-88.
4. Roberts, Sarah. (2023). *Navigating the Startup Ecosystem: Trends and Challenges*. New York: Springer.
5. Anderson, David. (2022). *Entrepreneurial Finance: Strategies for Startup Success*. Boston: Harvard Business Review Press.
6. Brown, Lisa. (2024). "Integrated Payment Solutions in Online Investment Platforms." *Journal of Financial Technology*, vol. 3, no. 4, pp. 112-128.
7. Patel, Rajesh. (2023). "Algorithmic Compatibility Assessment in Startup-Investor Matchmaking." *Journal of Financial Innovation and Technology*, vol. 6, no. 2, pp. 55-70.