UE22CS351A – DBMS

AUG - DEC 2025

Mini Project Report on

Startup and Investment Database System

Submitted to : Dr. Gamini Joshi Associate Professor

PES2UG23CS136 – Shreya Raj Boyapati
PES2UG23CS164– Deepthi J Kumbar

Department of CSE(AI&ML)
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

TABLE OF CONTENTS

1. INTRODUCTION

The Startup Investment Management System is a comprehensive web-based application designed to streamline and manage the complex relationships within the startup ecosystem. It enables efficient tracking of startups, founders, investors, funding rounds, mentors, and accelerators through an integrated, database-driven interface.This system allows users to add, edit, and manage key entities involved in startup investment processes while maintaining data consistency through stored procedures, triggers, and relational integrity. The intuitive Flask-based UI provides a centralized platform to visualize startup growth, monitor funding progress, and maintain professional investor and mentor networks — all within a single application.

2. PROBLEM DEFINITION

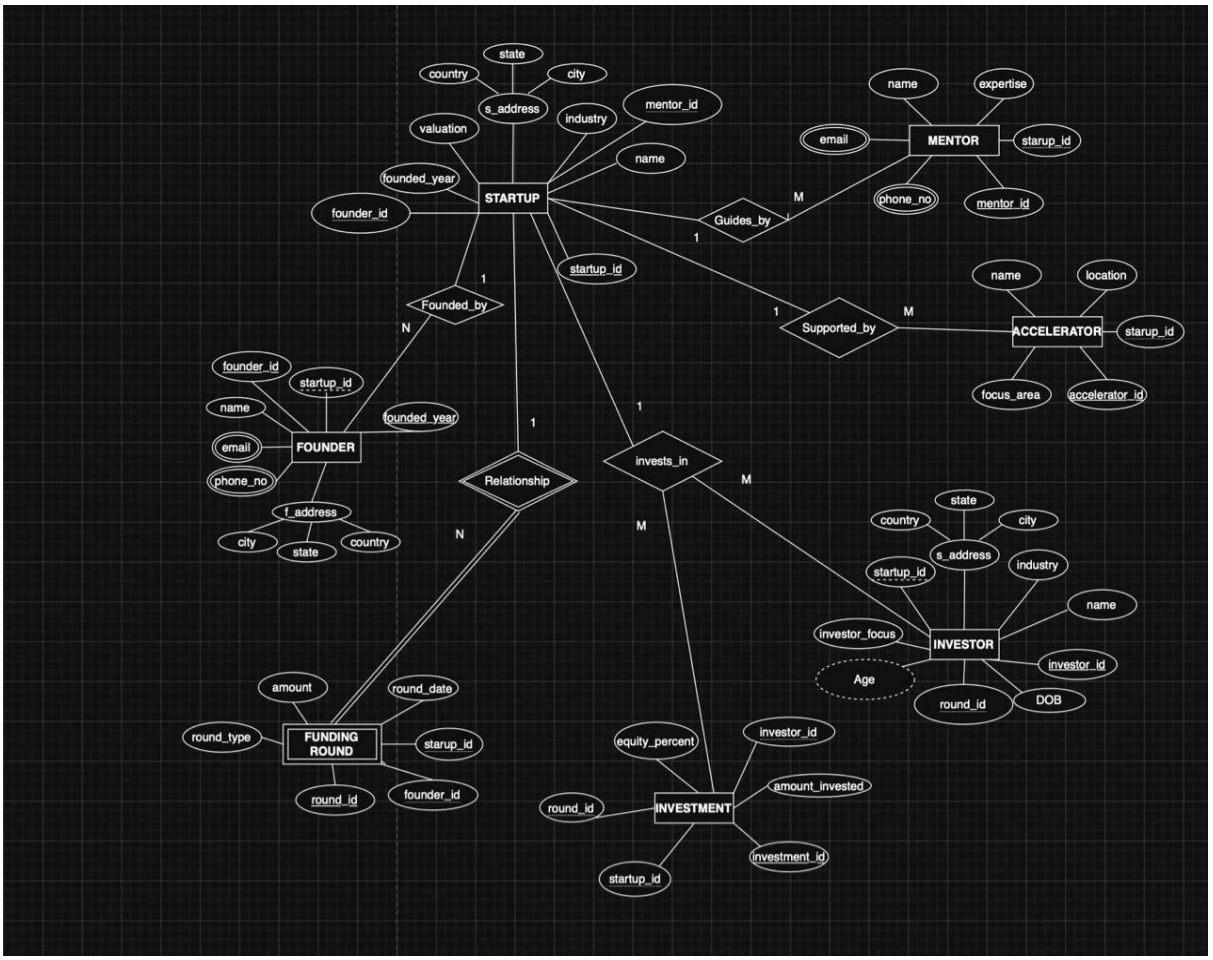
In the startup ecosystem, managing information related to founders, investors, funding rounds, and accelerators often becomes disorganized due to scattered data stored in multiple spreadsheets or manual records. This leads to challenges such as:

- Difficulty tracking relationships between startups and investors
- Inconsistent or duplicate data across different departments
- Lack of automated updates when related entities change (e.g., deleting a startup linked to multiple investors)
- Limited analytical visibility into funding patterns, mentorship involvement, or accelerator participation
- Furthermore, manual data handling is time-consuming, error-prone, and inefficient for organizations dealing with multiple startups and investors simultaneously.

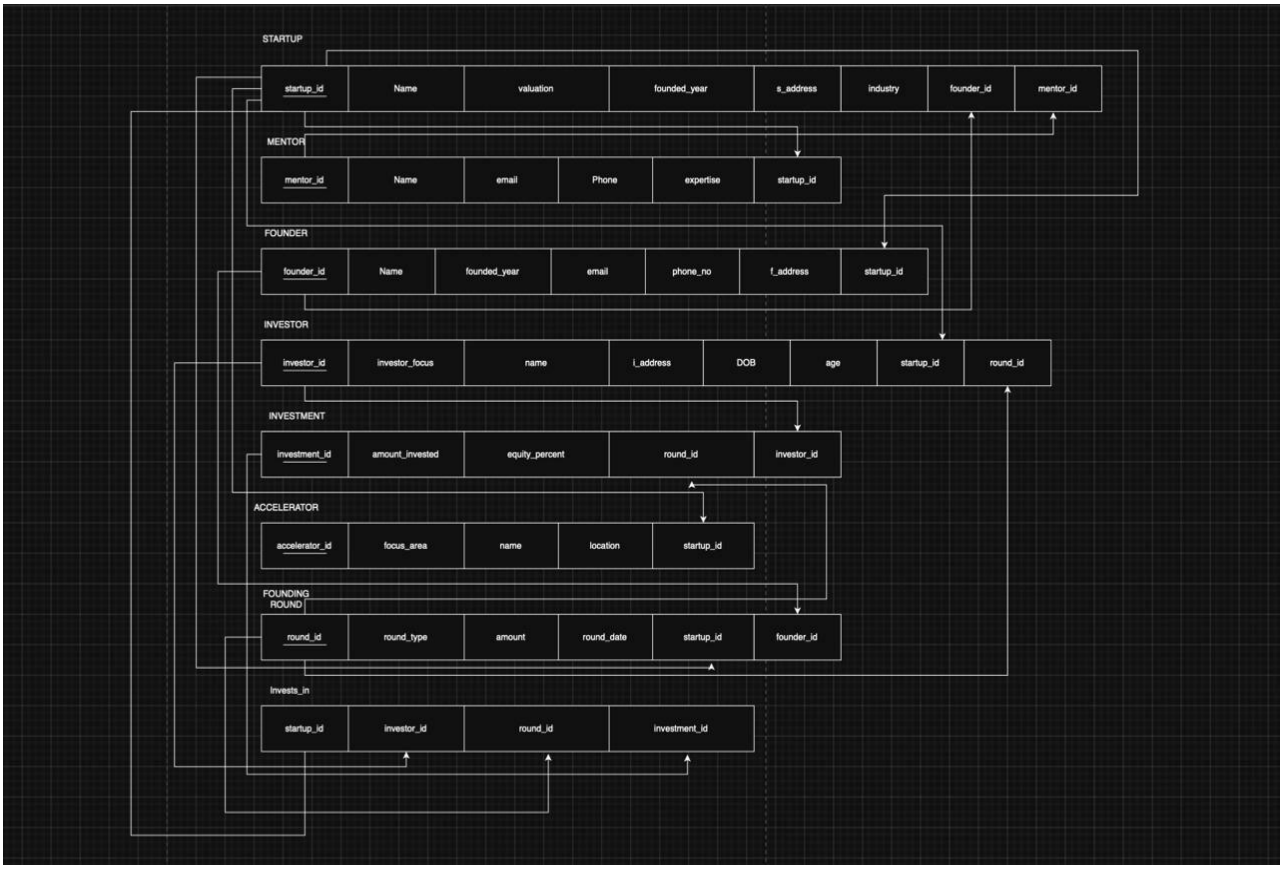
3. LIST OF SOFTWARES/TOOLS/PROGRAMMING LANGUAGES USED

- **SQL**
- **Python**
- **HTML**
- **Flask**

4. ER MODEL



5. ER TO RELATIONAL MAPPING



6. DDL STATEMENTS

```
CREATE DATABASE startup_investment_db;
USE startup_investment_db;
```

```
CREATE TABLE Founder (     founder_id INT PRIMARY
KEY AUTO_INCREMENT,     name VARCHAR(100)
NOT NULL,
   founded_year INT,
email VARCHAR(100),
f_address VARCHAR(255)
);

CREATE TABLE Mentor (     mentor_id INT PRIMARY
KEY AUTO_INCREMENT,     name VARCHAR(100)
NOT NULL,     email VARCHAR(100),     phone
VARCHAR(20),     expertise VARCHAR(100)
);

CREATE TABLE Startup (     startup_id INT PRIMARY
KEY AUTO_INCREMENT,     name VARCHAR(100)
NOT NULL,
   valuation DECIMAL(15,2),
founded_year INT,     s_address
VARCHAR(255),     industry
VARCHAR(100),
   founder_id INT,
mentor_id INT,
    FOREIGN KEY (founder_id) REFERENCES Founder(founder_id),
    FOREIGN KEY (mentor_id) REFERENCES Mentor(mentor_id)
);

CREATE TABLE Accelerator (     accelerator_id INT
PRIMARY KEY AUTO_INCREMENT,     name
VARCHAR(100) NOT NULL,     focus_area
VARCHAR(100),     location VARCHAR(100),
   startup_id INT,
```

```
    FOREIGN KEY (startup_id) REFERENCES Startup(startup_id)
);

CREATE TABLE Funding_Round (     round_id INT
PRIMARY KEY AUTO_INCREMENT,
    round_type VARCHAR(50),
amount DECIMAL(15,2),
round_date DATE,
    startup_id INT,
founder_id INT,
    FOREIGN KEY (startup_id) REFERENCES Startup(startup_id),
    FOREIGN KEY (founder_id) REFERENCES Founder(founder_id)
);


CREATE TABLE Investor (     investor_id INT PRIMARY
KEY AUTO_INCREMENT,     name VARCHAR(100)
NOT NULL,
    investor_focus VARCHAR(100),     l_address VARCHAR(255),     DOB DATE,
investor_age INT AS (TIMESTAMPDIFF(YEAR, DOB, CURDATE())) VIRTUAL,
    startup_id INT,
    round_id INT,
    FOREIGN KEY (startup_id) REFERENCES Startup(startup_id),
    FOREIGN KEY (round_id) REFERENCES Funding_Round(round_id)
);

CREATE TABLE Investment (     investment_id INT
PRIMARY KEY AUTO_INCREMENT,
    amount_invested DECIMAL(15,2),
equity_percent DECIMAL(5,2),
round_id INT,     investor_id INT,
    FOREIGN KEY (round_id) REFERENCES Funding_Round(round_id),
    FOREIGN KEY (investor_id) REFERENCES Investor(investor_id)
```

```
);

CREATE TABLE Invests_In (
   startup_id INT,
investor_id INT,     round_id
INT,     investment_id INT,
   PRIMARY KEY (startup_id, investor_id, round_id, investment_id),
   FOREIGN KEY (startup_id) REFERENCES Startup(startup_id),
   FOREIGN KEY (investor_id) REFERENCES Investor(investor_id),
   FOREIGN KEY (round_id) REFERENCES Funding_Round(round_id),
   FOREIGN KEY (investment_id) REFERENCES Investment(investment_id)
);
CREATE TABLE Founder_Phone (
   founder_id INT,
   phone_no VARCHAR(20),
   PRIMARY KEY (founder_id, phone_no),
   FOREIGN KEY (founder_id) REFERENCES Founder(founder_id)
);
CREATE TABLE Mentor_Phone (
   mentor_id INT,
   phone_no VARCHAR(20),
   PRIMARY KEY (mentor_id, phone_no),
   FOREIGN KEY (mentor_id) REFERENCES Mentor(mentor_id)
);
```

7.DML STATEMENTS

```
103     VALUES
104     ('Ramesh Kumar', 2015, 'ramesh.kumar@gmail.com', 'Bengaluru, Karnataka'),
105     ('Anita Sharma', 2018, 'anita.sharma@gmail.com', 'Hyderabad, Telangana'),
106     ('Kiran Patel', 2020, 'kiran.patel@gmail.com', 'Ahmedabad, Gujarat'),
107     ('Priya Verma', 2019, 'priya.verma@gmail.com', 'Chennai, Tamil Nadu');
108 *   select * from Founder;
```

100% 23:108

Result Grid Filter Rows: Search Edit: Export/Import:

founder_id	name	founded_year	email	phone_no	f_address
1	Ramesh Kumar	2015	ramesh.kumar@gmail.com	NULL	Bengaluru, Karnataka
2	Anita Sharma	2018	anita.sharma@gmail.com	NULL	Hyderabad, Telangana
3	Kiran Patel	2020	kiran.patel@gmail.com	NULL	Ahmedabad, Gujarat
4	Priya Verma	2018	priya.verma@gmail.com	NULL	Chennai, Tamil Nadu
18	Shreya Raj	2005	shreyaraj@gmail.com	NULL	pes university
NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Mentor (name, email, phone, expertise,mentor_id)
VALUES
('Dr. Rajesh Mehta', 'rajesh.mehta@gmail.com', '9876543210', 'AI & Machine Learning',1),
('Sneha Kapoor', 'sneha.kapoor@gmail.com', '9876500011', 'Marketing & Branding',2),
('Amit Tiwari', 'amit.tiwari@gmail.com', '9008765432', 'Finance & Investment',3);
```

100% 22:115

Result Grid Filter Rows: Search Edit: Export/Import:

mentor_id	name	email	phone	expertise
1	Dr. Rajesh Mehta	rajesh.mehta@gmail.com	9876543210	AI & Machine Learning
2	Sneha Kapoor	sneha.kapoor@gmail.com	9876500011	Marketing & Branding
3	Amit Tiwari	amit.tiwari@gmail.com	9008765432	Finance & Investment
NULL	NULL	NULL	NULL	NULL

```
118    VALUES
119    (1,'TechNova', 5000000, 2017, 'Koramangala, Bengaluru', 'Software', 1, 1),
120    (2,'AgroLink', 3500000, 2019, 'Jubilee Hills, Hyderabad', 'Agritech', 2, 2),
121    (3,'MediCarePlus', 4200000, 2018, 'Andheri, Mumbai', 'Healthcare', 3, 3),
122    (4,'EduSmart', 2500000, 2020, 'T. Nagar, Chennai', 'EdTech', 4, 2);
123 *  select * from Startup;
```

100% 23:123

Result Grid Filter Rows: Search Edit: Export/Import:

startup_id	name	valuation	founded_year	s_address	industry	founder_id	mentor_id
1	TechNova	5000000.00	2017	Koramangala, Bengaluru	Software	1	1
2	AgroLink	3500000.00	2019	Jubilee Hills, Hyderabad	Agritech	2	2
3	MediCarePlus	4200000.00	2018	Andheri, Mumbai	Healthcare	3	3
4	EduSmart	2500000.00	2020	T. Nagar, Chennai	EdTech	4	2
30	mango	1000000.00	1995	bengaluru	ai	2	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
126    VALUES
127    (1,'Seed', 800000, '2021-03-15', 1, 1),
128    (2,'Series A', 1500000, '2022-06-21', 2, 2),
129    (3,'Series B', 2500000, '2023-04-05', 3, 3),
130    (4,'Pre-Seed', 400000, '2020-09-10', 4, 4);
131    select * from Funding_Round;
```

100% 29:131

Result Grid Filter Rows: Search Edit: Export/Import:

round_id	round_type	amount	round_date	startup_id	founder_id
1	Seed	800000.00	2021-03-15	1	1
2	Series A	1500000.00	2022-06-21	2	2
3	Series B	2500000.00	2023-04-05	3	3
4	Pre-Seed	400000.00	2020-09-10	4	4
NULL	NULL	NULL	NULL	NULL	NULL

```
VALUES
(1,'VentureCraft Capital', 'Fintech & SaaS', 'Whitefield, Bengaluru', '1982-09-12', 1, 1),
(2,'NextGen Ventures', 'Healthcare', 'Banjara Hills, Hyderabad', '1975-11-30', 2, 2),
(3,'Skyline Partners', 'EdTech', 'Powai, Mumbai', '1988-02-22', 3, 3),
(4,'Vision Growth Fund', 'Agritech', 'Anna Nagar, Chennai', '1980-07-15', 4, 4);
select* from Investor;
```

100% 23:139

Result Grid Filter Rows: Search Edit: Export/Import:

investor_id	name	investor_focus	l_address	startup_id	round_id	investor_age	DOB
1	VentureCraft Capital	Fintech & SaaS	Whitefield, Bengaluru	1	1	43	1982-09-12
2	NextGen Ventures	Healthcare	Banjara Hills, Hyderabad	2	2	49	1975-11-30
3	Skyline Partners	EdTech	Powai, Mumbai	3	3	37	1988-02-22
4	Vision Growth Fund	Agritech	Anna Nagar, Chennai	4	4	45	1980-07-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Accelerator (name, focus_area, location, startup_id)
VALUES
('LaunchPad Labs', 'AI & Machine Learning', 'Bengaluru', 1),
('AgriBoost', 'Sustainable Agriculture', 'Hyderabad', 2),
('HealthXcelerate', 'HealthTech', 'Mumbai', 3),
('EduLift', 'Educational Technology', 'Chennai', 4);
```

100% 27:147

Result Grid Filter Rows: Search Edit: Export/Import:

accelerator_...	name	focus_area	location	startup_id
1	LaunchPad Labs	AI & Machine Learning	Bengaluru	1
2	AgriBoost	Sustainable Agriculture	Hyderabad	2
3	HealthXcelerate	HealthTech	Mumbai	3
4	EduLift	Educational Technology	Chennai	4
NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Investor_Phone (investor_id, phone_no)
VALUES
(1, '9876001234'),
(2, '9998801122'),
(3, '8887703344'),
(4, '7776612233');
```

100% 30:155

Result Grid Filter Rows: Search Edit: Export/Import:

investor_id	phone_no
1	9876001234
2	9998801122
3	8887703344
4	7776612233
NULL	NULL

8. QUERIES

8.1 SIMPLE QUERY WITH GROUP BY, AGGREGATE

```
@app.route('/investors') def
investors():
    cursor.execute("""
        SELECT i.*, s.name AS startup_name, f.round_type AS round_name
        FROM Investor i
        LEFT JOIN Startup s ON i.startup_id = s.startup_id
        LEFT JOIN Funding_Round f ON i.round_id = f.round_id
        ORDER BY s.name
    """)     investors = cursor.fetchall()     return
render_template('investors.html', investors=investors)
```

8.2 UPDATE OPERATION

```
@app.route('/edit_startup/<int:startup_id>', methods=['POST'])
def edit_startup(startup_id):     name = request.form['name']
valuation = request.form['valuation']     founded_year =
request.form['founded_year']     s_address =
request.form['s_address']     industry = request.form['industry']
founder_id = request.form['founder_id']

    mentor_id = request.form['mentor_id']


    cursor.execute("""
```

```
        UPDATE Startup
        SET name=%s, valuation=%s, founded_year=%s, s_address=%s,
industry=%s, founder_id=%s, mentor_id=%s
        WHERE startup_id=%s
    """, (name, valuation, founded_year, s_address, industry, founder_id,
mentor_id, startup_id))     db.commit()     return redirect('/startups')
```

8.3 DELETE OPERATION

```
@app.route('/delete_startup/<int:startup_id>') def
delete_startup(startup_id):
    try:
        cursor.execute("DELETE FROM Investor_Phone WHERE investor_id IN
(SELECT investor_id FROM Investor WHERE startup_id=%s)", (startup_id,))
        cursor.execute("DELETE FROM Investor WHERE startup_id=%s",
(startup_id,))
        cursor.execute("DELETE FROM Funding_Round WHERE startup_id=%s",
(startup_id,))
        cursor.execute("DELETE FROM Accelerator WHERE startup_id=%s",
(startup_id,))
        cursor.execute("DELETE FROM Startup WHERE startup_id=%s",
(startup_id,))
        db.commit()         print(f"☑ Startup {startup_id}
deleted successfully!")     except mysql.connector.Error as
err:
        db.rollback()         print(f"❌ Error
deleting startup: {err}")     return
redirect('/startups')
```

8.4 CORRELATED QUERY

```
@app.route('/get_investor_age', methods=['POST'])
def get_investor_age():     investor_id =
request.form['investor_id']     cursor.execute("""
SELECT name,
            YEAR(CURDATE()) - YEAR(DOB) -
            (DATE_FORMAT(CURDATE(), '%m%d') < DATE_FORMAT(DOB,
'%m%d')) AS Age
        FROM Investor
        WHERE investor_id = %s
    """, (investor_id,))     age = cursor.fetchone()     return
render_template('functions_investor.html', age=age)
```

8.5 FUNCTIONAL QUERY

```
@app.route('/get_startup_age', methods=['POST'])
def get_startup_age():     startup_id =
request.form['startup_id']
    cursor.execute("""
        SELECT name, (YEAR(CURDATE()) - founded_year) AS Age
        FROM Startup
        WHERE startup_id = %s
    """, (startup_id,))
data = cursor.fetchone()
    return render_template('functions_startup.html', data=data)
```

8.6 AGGREGATE QUERY WITH GROUPING

```
@app.route('/funding_summary') def
funding_summary():
cursor.execute("""
        SELECT s.name AS Startup_Name, COUNT(f.round_id) AS
Total_Rounds,
            SUM(f.amount) AS Total_Funding
```

```
    FROM Funding_Round f
    JOIN Startup s ON f.startup_id = s.startup_id
    GROUP BY s.startup_id
    ORDER BY Total_Funding DESC;
"""")
summary = cursor.fetchall()
return render_template('funding_summary.html', summary=summary)
```

8.7 Nested Queries

1. Find all startups whose valuation is above the average valuation

---

```
SELECT name, valuation
FROM Startup
WHERE valuation > (
   SELECT AVG(valuation)
   FROM Startup
);
```

2. List founders who have more than one startup ----------

```
SELECT name
FROM Founder
WHERE founder_id IN (
   SELECT founder_id
   FROM Startup
   GROUP BY founder_id
   HAVING COUNT(*) > 1
);
```

3️⃣ Find investors who invested in startups located in Bengaluru

---

```
SELECT name
FROM Investor
WHERE startup_id IN (
    SELECT startup_id
    FROM Startup
    WHERE s_address LIKE '%Bengaluru%'
);
```

4️⃣ Get startups whose founder email ends with '@gmail.com'

```
SELECT name
FROM Startup
WHERE founder_id IN (
    SELECT founder_id
    FROM Founder
    WHERE email LIKE '%@gmail.com'
);
```

5️⃣ Find founders who have NOT started any startup

```
SELECT name
```

```
FROM Founder
WHERE founder_id NOT IN (
    SELECT founder_id
    FROM Startup
);
```

6️⃣ Get investors who invested in the highest-amount funding round

```
-----------------------------------------------------------------
SELECT name
FROM Investor
WHERE round_id IN (
    SELECT round_id
    FROM Funding_Round
    WHERE amount = (
        SELECT MAX(amount)
        FROM Funding_Round
    )
);
```

7️⃣ List startups mentored by mentors with "AI" expertise

```
-----------------------------------------------------------
SELECT name
FROM Startup
WHERE mentor_id IN (
    SELECT mentor_id
```

```
    FROM Mentor
    WHERE expertise LIKE '%AI%'
);
```

8️⃣ Get founders whose phone number contains '99'

```
-----------------------------------------------
SELECT name
FROM Founder
WHERE founder_id IN (
    SELECT founder_id
    FROM Founder_Phone
    WHERE phone_no LIKE '%99%'
);
```

9️⃣ Show investors older than average investor age

```
-----------------------------------------------
SELECT name, DOB
FROM Investor
WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) > (
    SELECT AVG(TIMESTAMPDIFF(YEAR, DOB, CURDATE()))
    FROM Investor
);
```

🔟 Startups whose founders have phone numbers starting with '9'

```
SELECT name
FROM Startup
WHERE founder_id IN (
    SELECT founder_id
    FROM Founder_Phone
    WHERE phone_no LIKE '9%'
```

9. STORED PROCEDURES, FUNCTIONS AND TRIGGERS

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `add_startup_with_founder`(
  IN p_startup_name VARCHAR(100),
  IN p_industry VARCHAR(100),
  IN p_founded_year INT,
  IN p_valuation DECIMAL(15,2),
  IN p_city VARCHAR(100),
  IN p_state VARCHAR(100),
  IN p_country VARCHAR(100),
  IN p_founder_name VARCHAR(100),
  IN p_founder_email VARCHAR(100),
  IN p_founder_phone VARCHAR(15)
)
BEGIN
  DECLARE new_founder_id INT;

  INSERT INTO FOUNDER(name, email, phone_no, city, state, country, founded_year)
VALUES (p_founder_name, p_founder_email, p_founder_phone, p_city, p_state, p_country,
p_founded_year);
```

```
  SET new_founder_id = LAST_INSERT_ID();

  INSERT INTO STARTUP(name, industry, founded_year, valuation, city, state, country,
founder_id)
  VALUES (p_startup_name, p_industry, p_founded_year, p_valuation, p_city, p_state,
p_country, new_founder_id);
END
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_total_funding`(IN p_startup_id
INT)
BEGIN
  SELECT s.name AS startup_name, SUM(f.amount) AS total_funding
  FROM STARTUP s
  JOIN FUNDING_ROUND f ON s.startup_id = f.startup_id
  WHERE s.startup_id = p_startup_id
  GROUP BY s.startup_id;
END
```

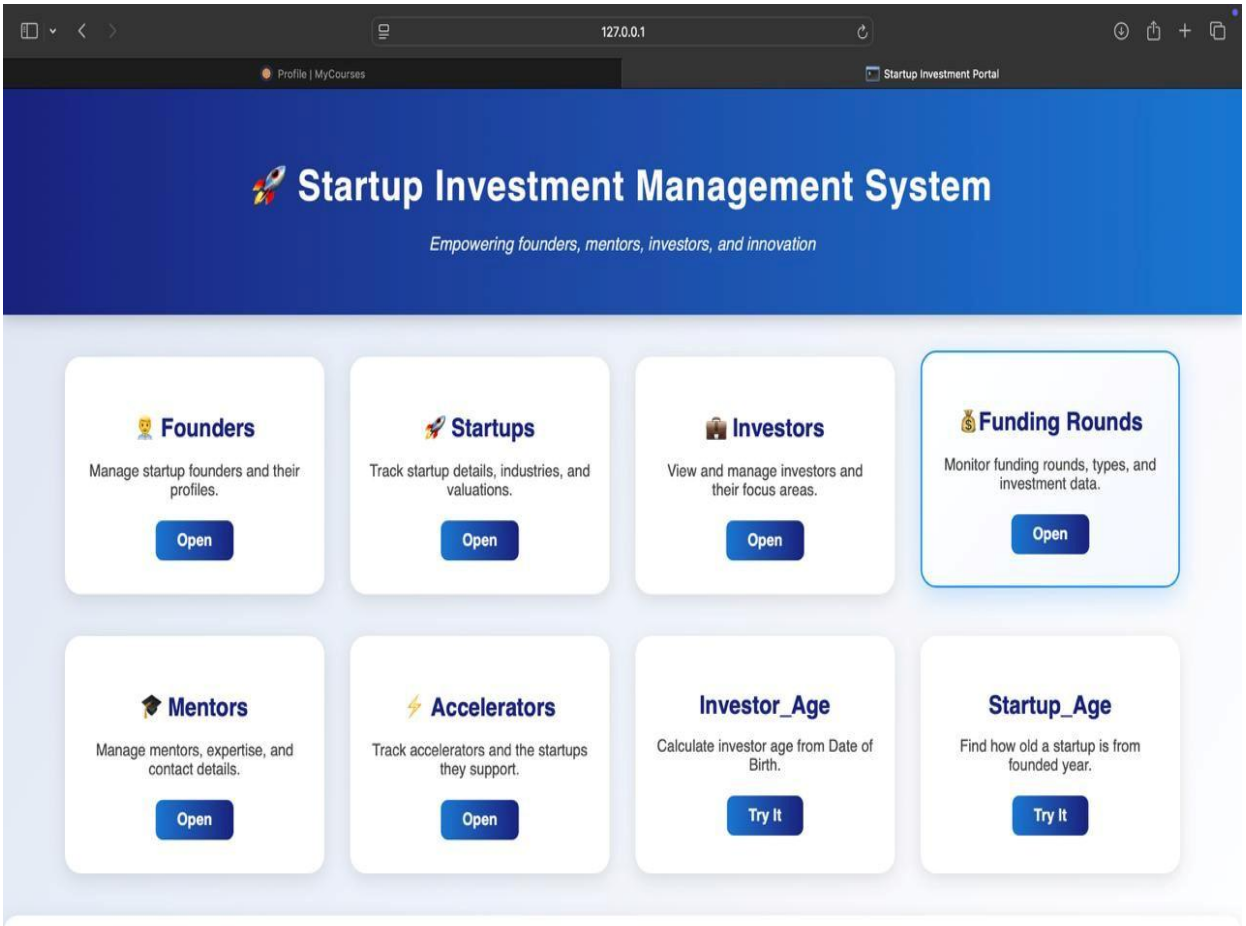
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `record_investment`(
  IN p_investor_id INT,
  IN p_startup_id INT,
  IN p_round_id INT,
  IN p_amount DECIMAL(15,2),
  IN p_equity DECIMAL(5,2)
```

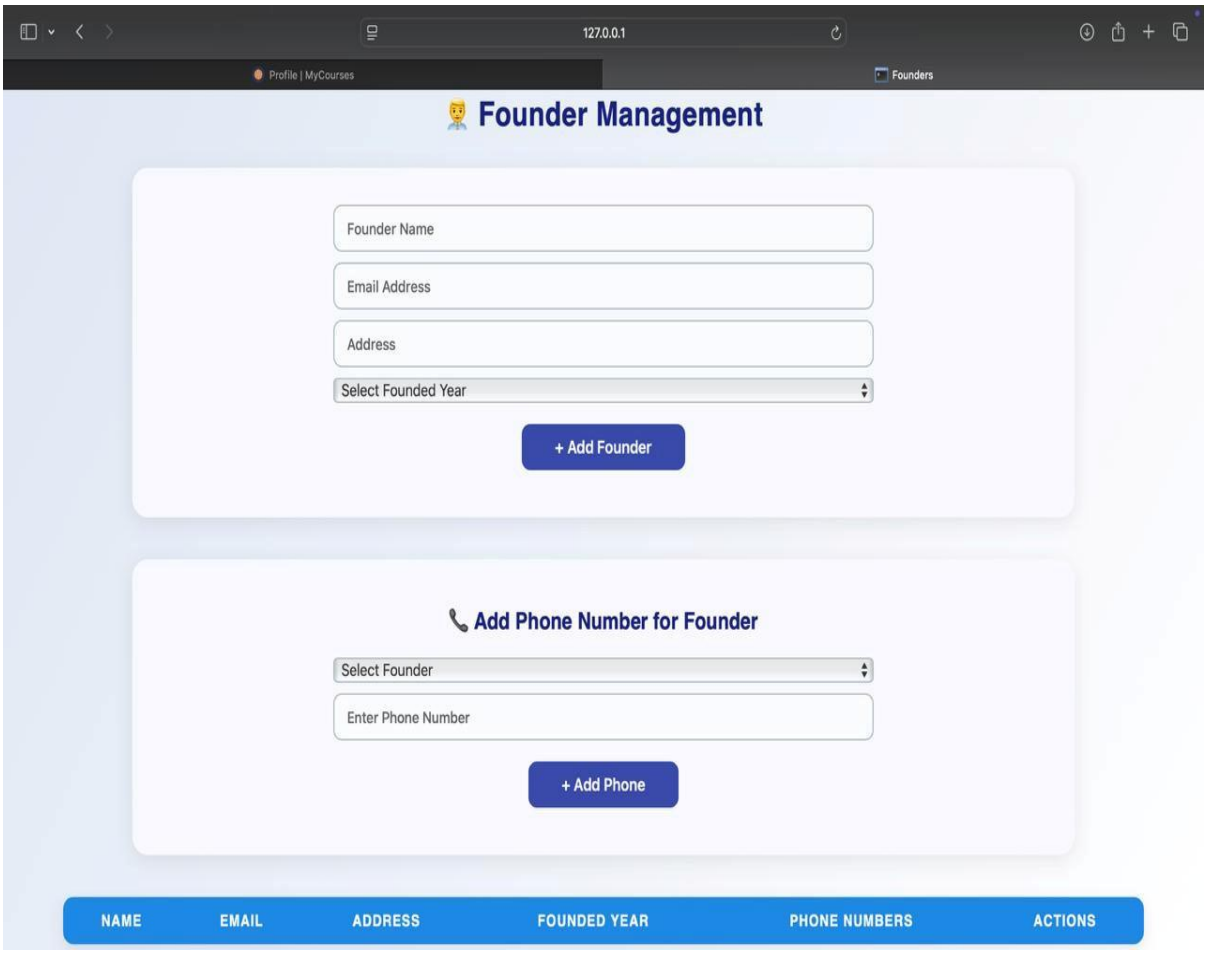
```
)
BEGIN
  INSERT INTO INVESTMENT(investor_id, startup_id, round_id, amount_invested,
equity_percent)
  VALUES (p_investor_id, p_startup_id, p_round_id, p_amount, p_equity);

  UPDATE FUNDING_ROUND
  SET amount = amount + p_amount
  WHERE round_id = p_round_id;
END
```

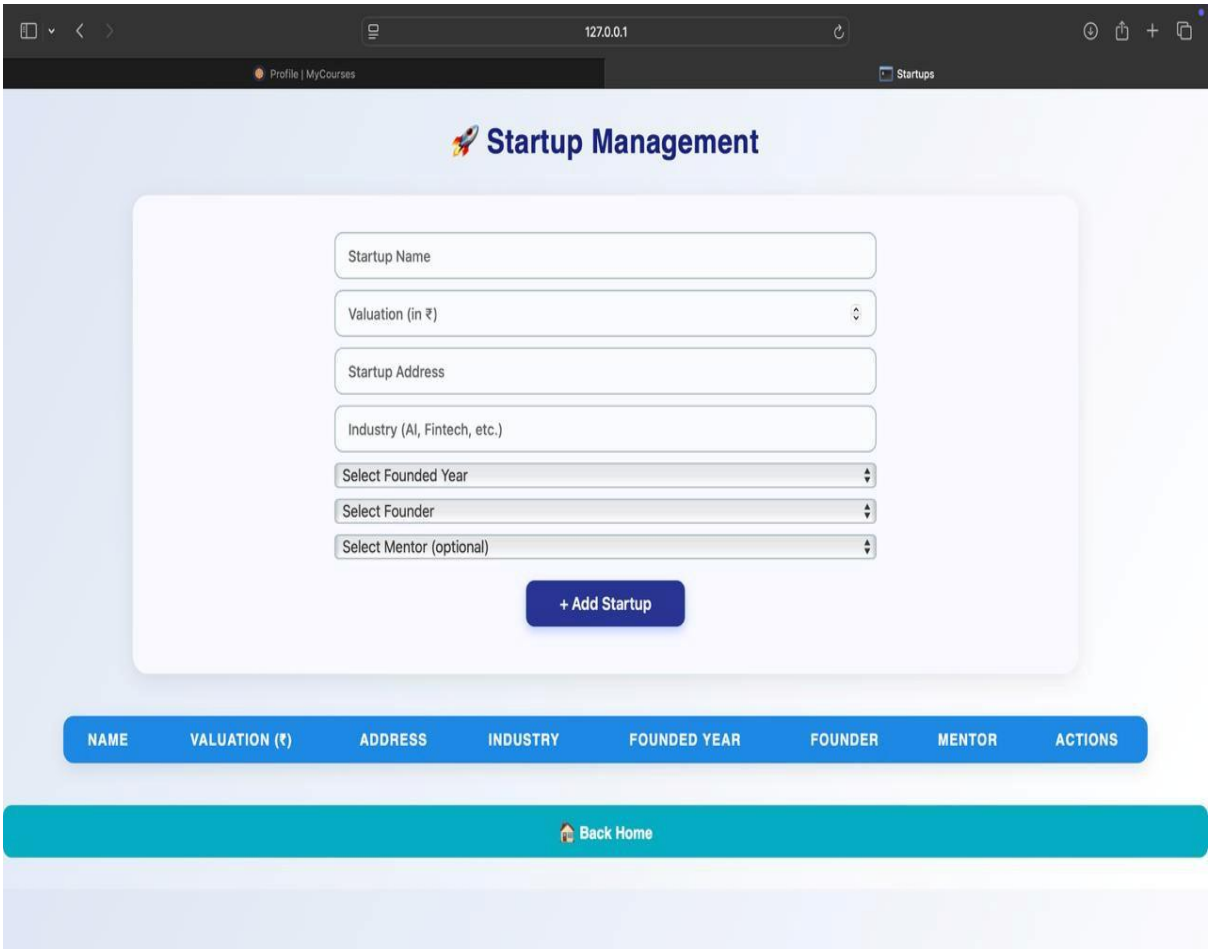
```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_investor_age`(dob DATE)
RETURNS int
    DETERMINISTIC
BEGIN
  RETURN TIMESTAMPDIFF(YEAR, dob, CURDATE());
END
```

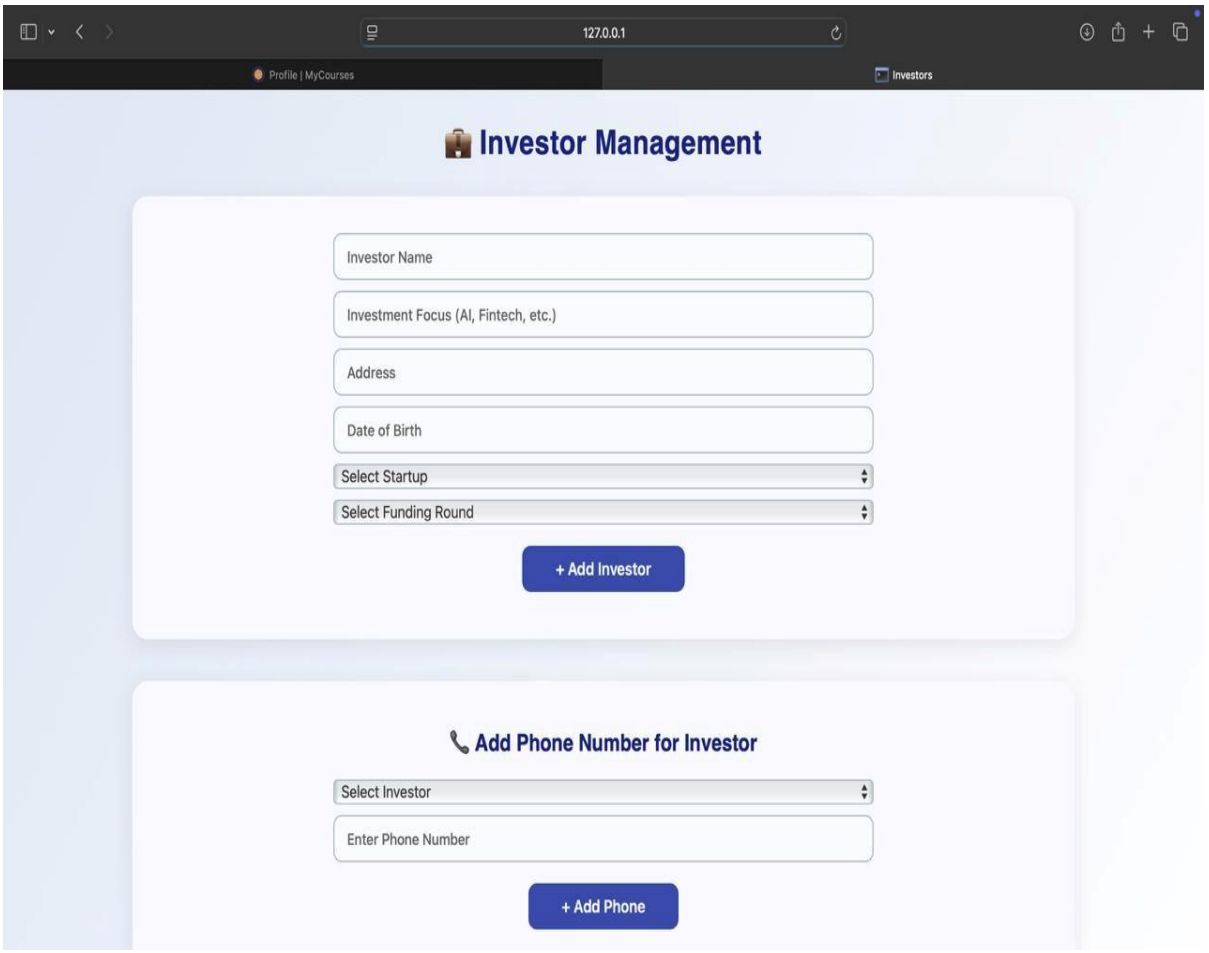
```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_startup_age`(founded_year INT)
RETURNS int
    DETERMINISTIC
BEGIN
  RETURN YEAR(CURDATE()) - founded_year;
END
```

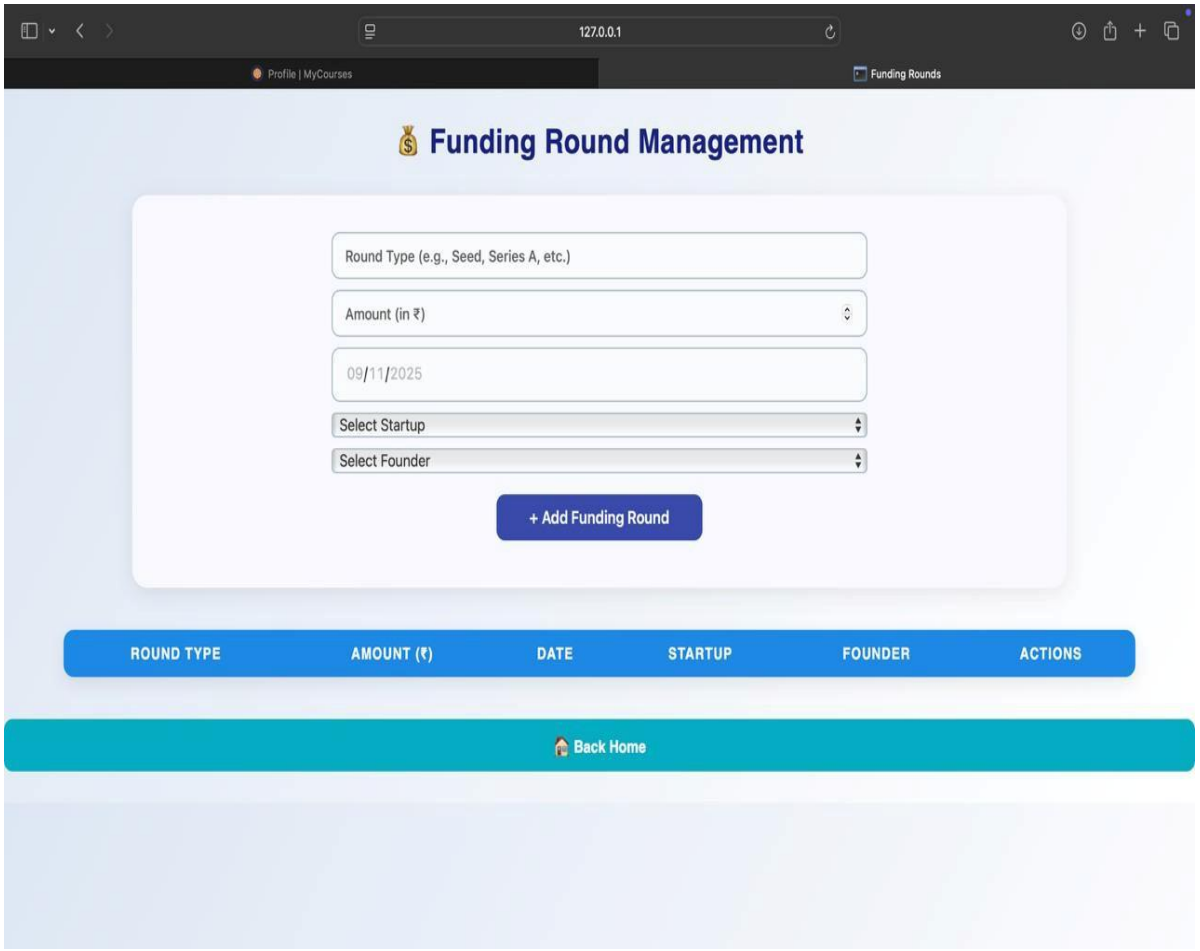
10. FRONT END DEVELOPMENT

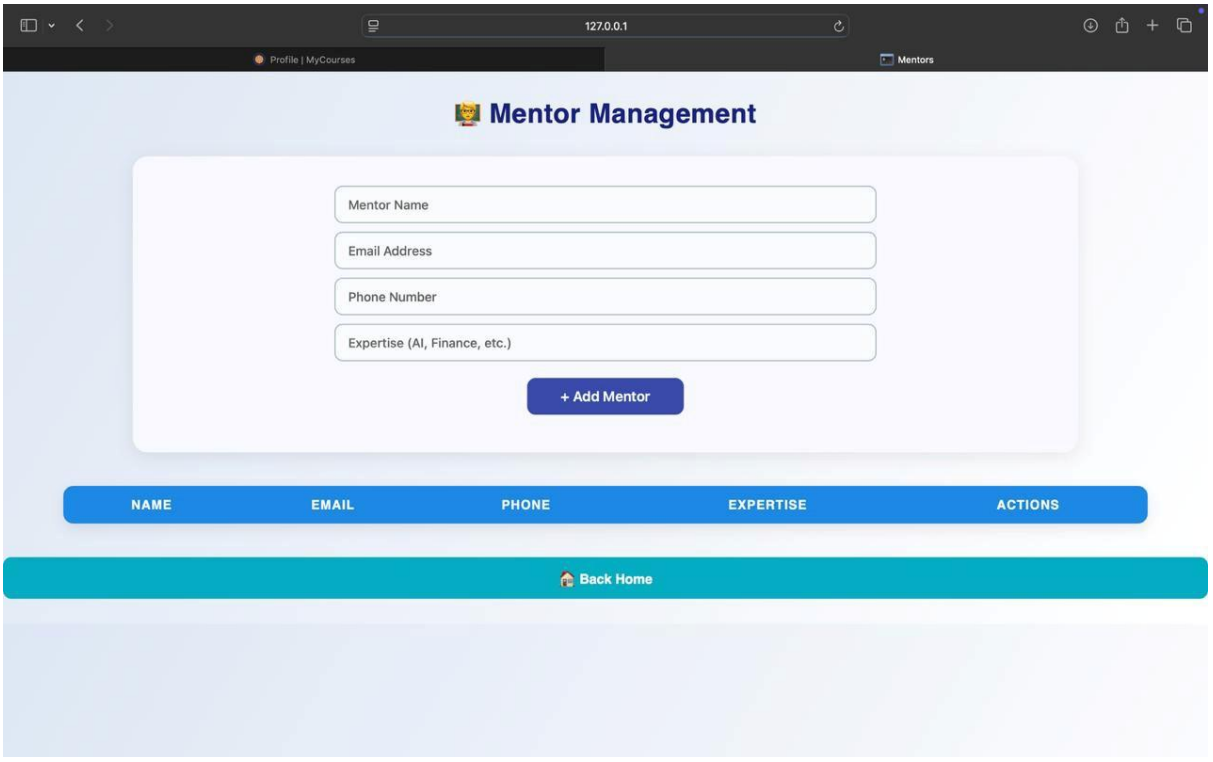


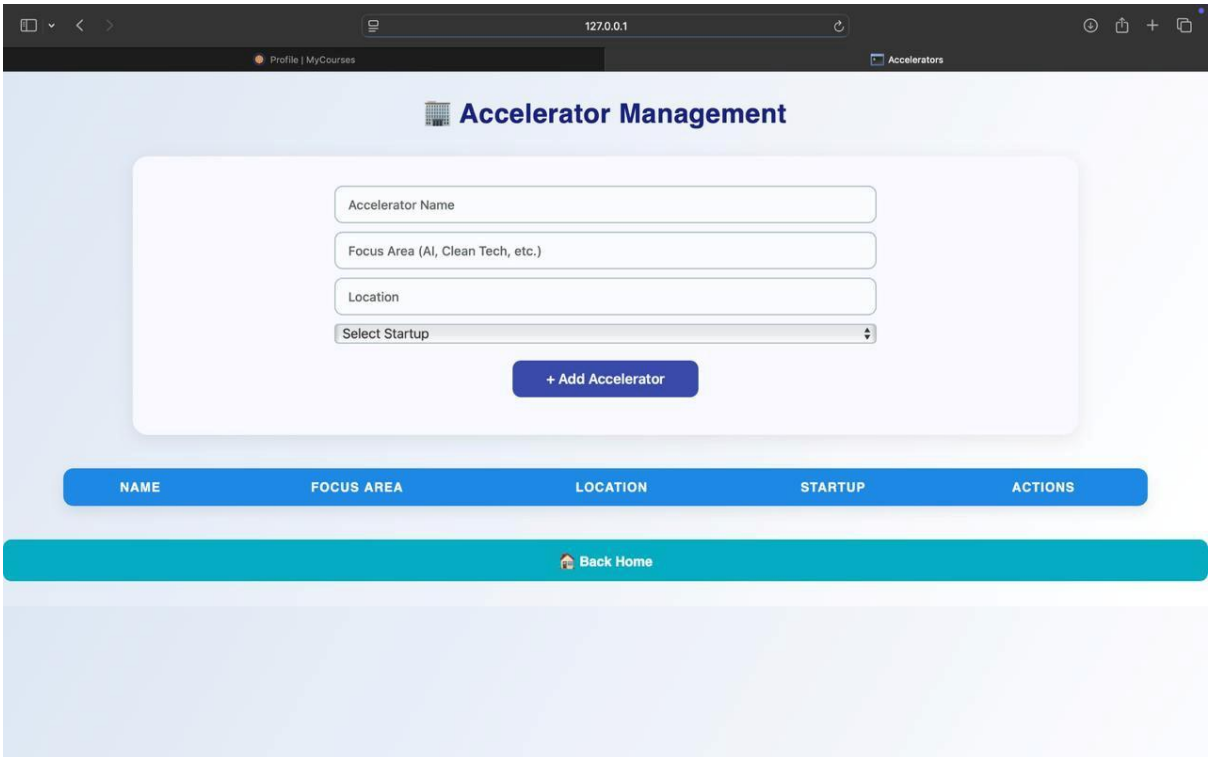

127.0.0.1
Profile | MyCourses
Founders
Founder Management
Founder Name
Email Address
Address
Select Founded Year
+ Add Founder
Add Phone Number for Founder
Select Founder
Enter Phone Number
+ Add Phone
NAME
EMAIL
ADDRESS
FOUNDED YEAR
PHONE NUMBERS
ACTIONS

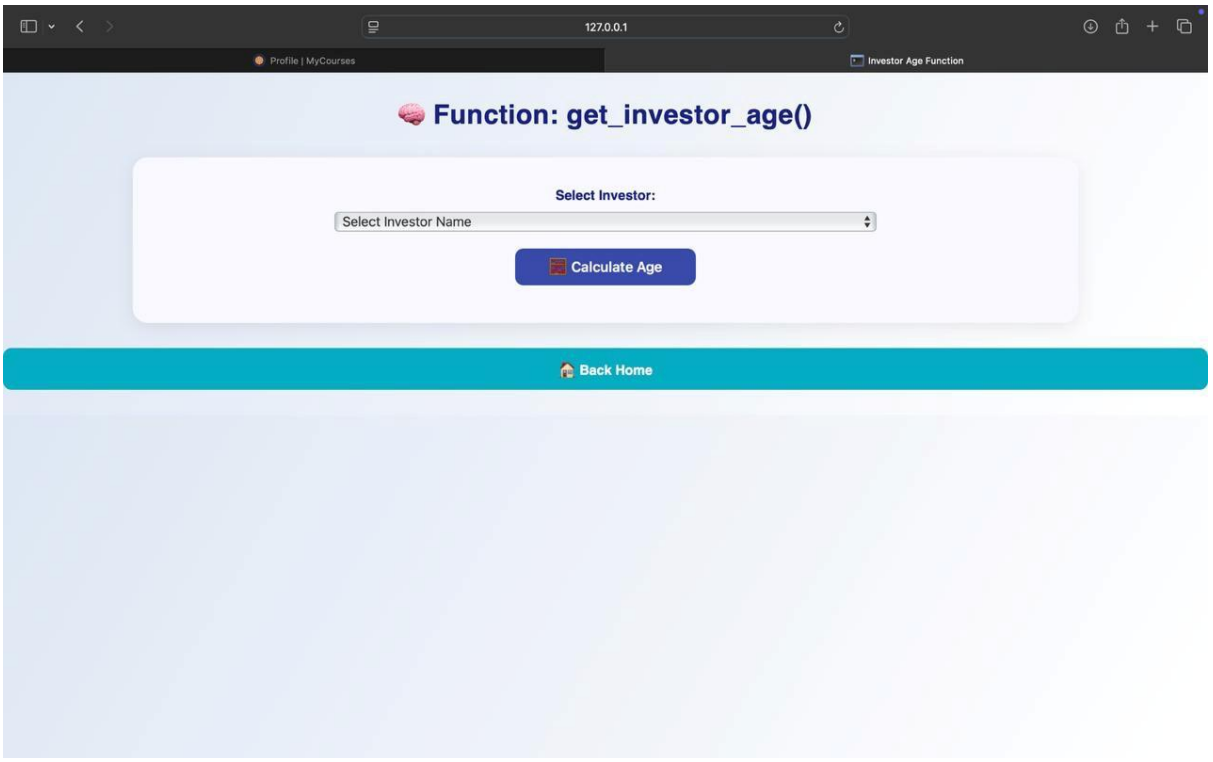

127.0.0.1
Profile | MyCourses
Startups
Startup Management
Startup Name
Valuation (in ₹)
Startup Address
Industry (AI, Fintech, etc.)
Select Founded Year
Select Founder
Select Mentor (optional)
+ Add Startup
NAME
VALUATION (₹)
ADDRESS
INDUSTRY
FOUNDED YEAR
FOUNDER
MENTOR
ACTIONS
Back Home

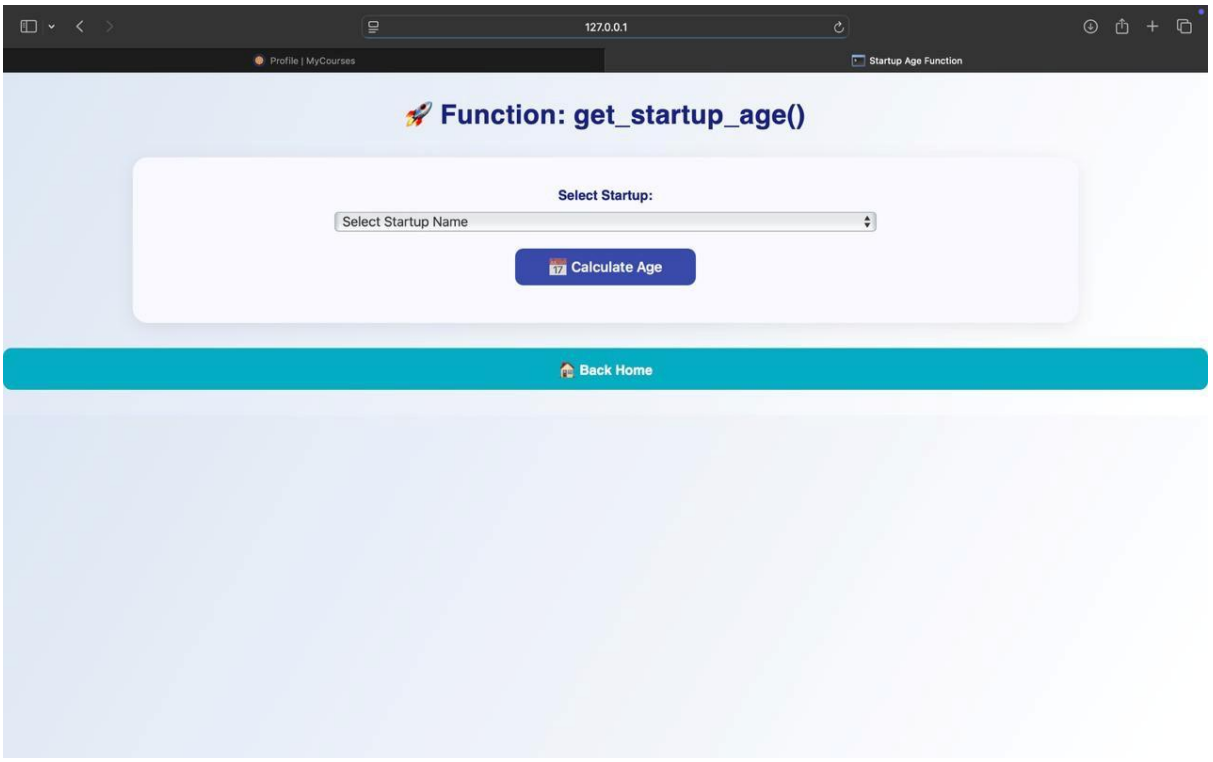

127.0.0.1
Profile | MyCourses
Investors
Investor Management
Investor Name
Investment Focus (AI, Fintech, etc.)
Address
Date of Birth
Select Startup
Select Funding Round
+ Add Investor
Add Phone Number for Investor
Select Investor
Enter Phone Number
+ Add Phone

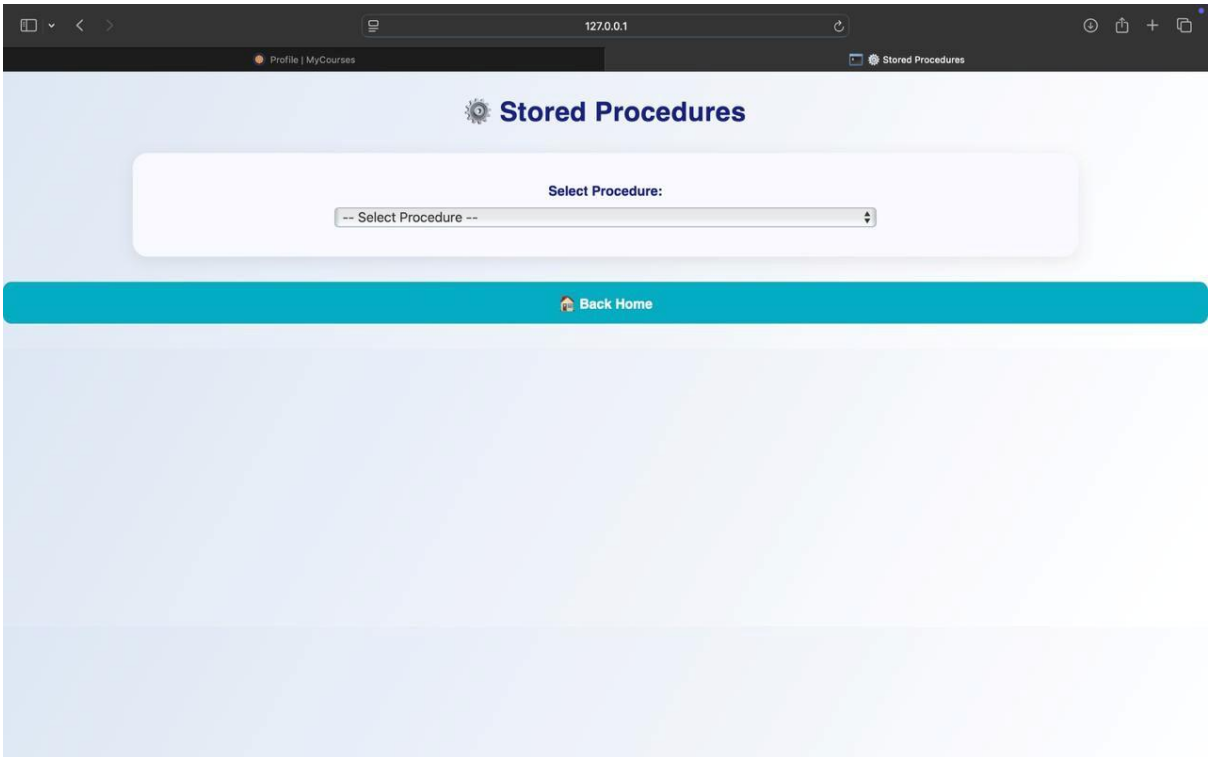

127.0.0.1
Profile | MyCourses
Funding Rounds
Funding Round Management
Round Type (e.g., Seed, Series A, etc.)
Amount (in ₹)
09/11/2025
Select Startup
Select Founder
+ Add Funding Round
ROUND TYPE
AMOUNT (₹)
DATE
STARTUP
FOUNDER
ACTIONS
Back Home


127.0.0.1
Profile | MyCourses
Mentors
Mentor Management
Mentor Name
Email Address
Phone Number
Expertise (AI, Finance, etc.)
+ Add Mentor
NAME
EMAIL
PHONE
EXPERTISE
ACTIONS
Back Home


127.0.0.1
Profile | MyCourses
Accelerators
Accelerator Management
Accelerator Name
Focus Area (AI, Clean Tech, etc.)
Location
Select Startup
+ Add Accelerator
NAME
FOCUS AREA
LOCATION
STARTUP
ACTIONS
Back Home


127.0.0.1
Profile | MyCourses
Investor Age Function
Function: get_investor_age()
Select Investor:
Select Investor Name
Calculate Age
Back Home





11. GITHUB REPO LINK

https://github.com/pes2ug23cs136/PES2UG23CS136_Startup-Investor-Database_DBMSminiproject

12. SQL

It is Uplaoded separately.

13. REFERENCES

MySQL Documentation – https://dev.mysql.com/doc/

Flask Documentation – https://flask.palletsprojects.com/

W3Schools SQL Tutorial – https://www.w3schools.com/sql/

Python Official Documentation – https://docs.python.org/3/

GeeksforGeeks – Tutorials on DBMS, SQL, and Flask Stack Overflow – Community discussions on Flask and MySQL integration Bootstrap Documentation – https://getbootstrap.com/

dotenv (Python) Library – https://pypi.org/project/python-dotenv/

Werkzeug Security Library – https://werkzeug.palletsprojects.com/