Artificial Intelligence & Machine Learning

Project Documentation

1. Introduction

- **Project Title:** Prosperity Prognosticator
- **Team Members: 4**

Prosperity Prognosticator is an Artificial Intelligence and Machine Learning–based application designed to predict the future outcome of startups. The system helps investors and startup founders make informed decisions by analyzing historical startup data and predicting whether a startup is likely to be **Acquired** or **Closed**.

2. Project Overview

Purpose

The main purpose of this project is to reduce uncertainty in startup investment decisions by providing a data-driven prediction system. By using machine learning algorithms, the application analyzes startup-related parameters and predicts outcomes with high accuracy.

Goals

- To apply AI and ML techniques to a real-world business problem
- To assist investors and founders in decision-making
- To build a scalable and user-friendly web application

Key Features

- Startup success prediction using machine learning
- User-friendly web interface
- Real-time prediction results
- Secure authentication system
- Scalable architecture

3. Architecture

Frontend Architecture (React)

The frontend is developed using **React**, which provides a responsive and dynamic user interface. React components handle user input, form validation, and display prediction results. The frontend communicates with the backend through RESTful APIs.

Backend Architecture (Node.js & Express.js)

The backend is built using **Node.js** and **Express.js**. It handles API requests, user authentication, data processing, and communication with the machine learning model. Express.js ensures structured routing and middleware support.

Database Architecture (MongoDB)

MongoDB is used as a NoSQL database to store user information, startup input data, and prediction logs. The backend interacts with MongoDB using schemas and models to perform CRUD operations efficiently.

4. Setup Instructions

Prerequisites

- Node.js (v14 or above)
- MongoDB
- npm (Node Package Manager)
- Git

Installation Steps

1. Clone the repository:
2. git clone <repository-url>
3. Navigate to the project folder.
4. Install frontend dependencies:
5. cd client
6. npm install
7. Install backend dependencies:
8. cd server
9. npm install
10. Configure environment variables (MongoDB URI, JWT secret, etc.).

5. Folder Structure

Client (Frontend)

```
client/
├── src/
│   ├── components/
│   ├── pages/
│   ├── services/
│   ├── App.js
│   └── index.js
│
```

- Components handle UI elements
- Pages represent different screens
- Services manage API calls

Server (Backend)

```
server/
├── controllers/
├── routes/
├── models/
├── middleware/
├── app.js
└── server.js
```

- Controllers contain business logic
- Routes define API endpoints
- Models define database schemas

6. Running the Application

Start Frontend

```
cd client
npm start
```

Start Backend

```
cd server
npm start
```

The application will run locally and can be accessed through a web browser.

7. API Documentation

Authentication APIs

- POST /api/register – Register a new user
- POST /api/login – Login user

Prediction API

- POST /api/predict
    - Input: Startup details (funding, milestones, location, etc.)
    - Output: Prediction result (Acquired / Closed)

8. Authentication

Authentication is handled using **JWT (JSON Web Tokens)**.

- Users authenticate using email and password
- A token is generated after successful login
- Protected routes require a valid token

This ensures secure access and authorization.

9. User Interface

The user interface is simple and intuitive. It includes:

- Login and Registration pages
- Startup data input form
- Prediction result dashboard

Screenshots and UI previews are included in the documentation.

10. Testing

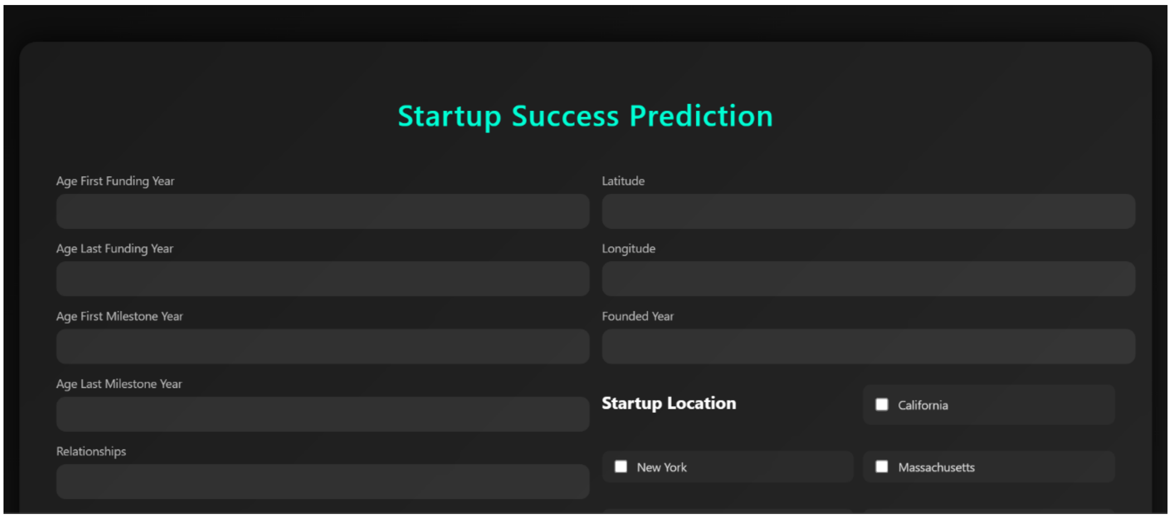
Testing Strategy

- Unit testing for backend APIs
- Manual testing for frontend UI
- Model performance testing using accuracy metrics

Tools Used

- Postman (API testing)
- Manual UI testing
- Model evaluation metrics

11. Screenshots / Demo


Startup Success Prediction

Age First Funding Year

Age Last Funding Year

Age First Milestone Year

Age Last Milestone Year

Relationships

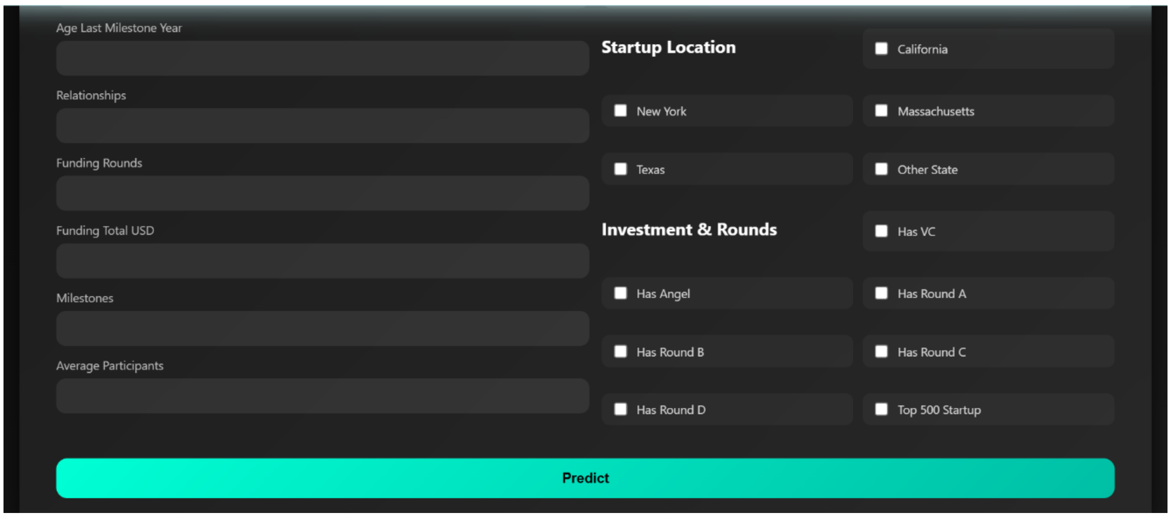
Latitude

Longitude

Founded Year

Startup Location

- [ ] California
- [ ] New York
- [ ] Massachusetts


Age Last Milestone Year

Relationships

Funding Rounds

Funding Total USD

Milestones

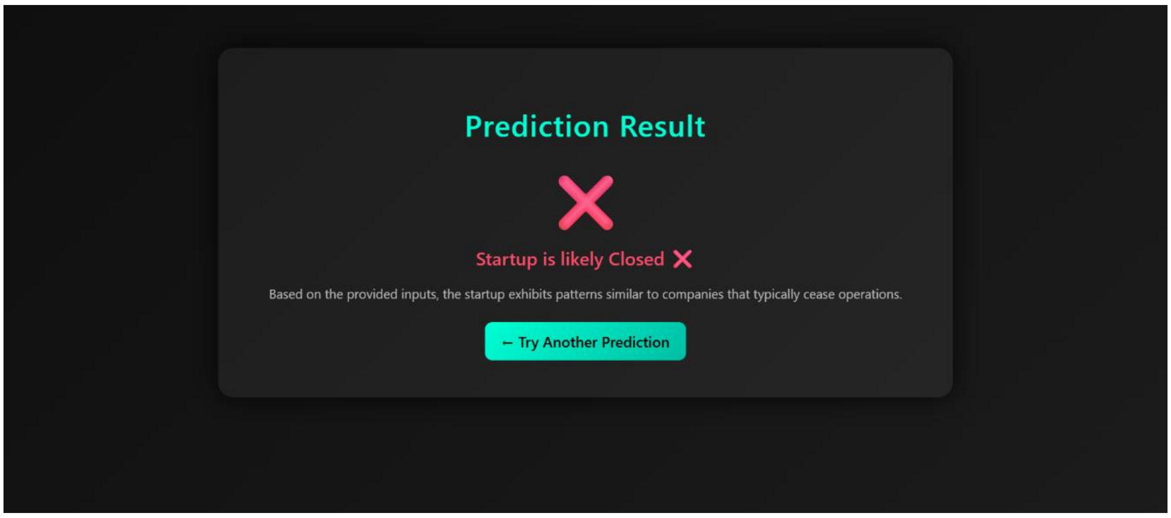
Average Participants

Startup Location

- [ ] California
- [ ] New York
- [ ] Massachusetts
- [ ] Texas
- [ ] Other State

Investment & Rounds

- [ ] Has VC
- [ ] Has Angel
- [ ] Has Round A
- [ ] Has Round B
- [ ] Has Round C
- [ ] Has Round D
- [ ] Top 500 Startup


Prediction Result

Startup is likely Closed ✗

Based on the provided inputs, the startup exhibits patterns similar to companies that typically cease operations.

A demo video or live demo link can be provided to showcase the application.
Startup Prediction - Google Chrome 2026-02-18 17-12-00.mp4

12. Known Issues

- Prediction accuracy depends on data quality
- Performance may vary with very large datasets
- Limited real-time data integration

13. Future Enhancements

- Deploy application on cloud platforms
- Improve prediction accuracy using advanced models
- Add real-time startup data sources
- Enhance UI using advanced visualization libraries
- Mobile application support