FINAL PROJECT REPORT

Date	17 February 2026
Team ID	LTVIP2026TMIDS36980
Project Name	Gemini Pro Financial Decoder
Maximum Marks	4 Marks

1. INTRODUCTION

1.1 Project Overview

The project titled “Gemini Pro Financial Decoder” is a web-based application that simplifies the analysis of financial statements using Generative Artificial Intelligence. Financial statements such as Balance Sheets, Profit and Loss Statements, and Cash Flow Statements contain complex numerical data and technical terminology that are difficult to interpret for non-finance users.

This project enables users to upload financial data in CSV or Excel format and automatically generates simplified summaries and visual insights. The system uses Google Gemini AI to produce intelligent explanations and includes a fallback rule-based summary mechanism to handle AI quota limitations, ensuring uninterrupted functionality.

1.2 Purpose

The purpose of this project is to:

- Simplify financial statement analysis for non-finance users
- Automate interpretation of financial data using AI
- Reduce dependency on financial experts
- Provide clear summaries and visual representations
- Support informed and faster decision-making

2.IDEATION PHASE

2.1 Problem Statement

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Introduction

In today’s digital era, organizations generate a large amount of financial data. Analyzing financial statements such as Balance Sheets, Profit and Loss Statements, and Cash Flow Statements requires financial expertise and significant time. Small businesses, students, and non-finance professionals often find it difficult to interpret these documents accurately.

The Ideation Phase focuses on identifying this problem and proposing an innovative solution using modern technologies such as Generative Artificial Intelligence.

Problem Identification

Traditional financial analysis is:

- Time-consuming
- Dependent on financial experts
- Difficult for beginners
- Error-prone when done manually

There is a strong need for a system that can automatically analyse financial data and provide understandable insights.

Brainstorming Process

Multiple ideas were generated during brainstorming:

- Excel-based financial calculators
- Rule-based financial analysis software
- Dashboard-only visualization tools
- AI-powered financial analysis system

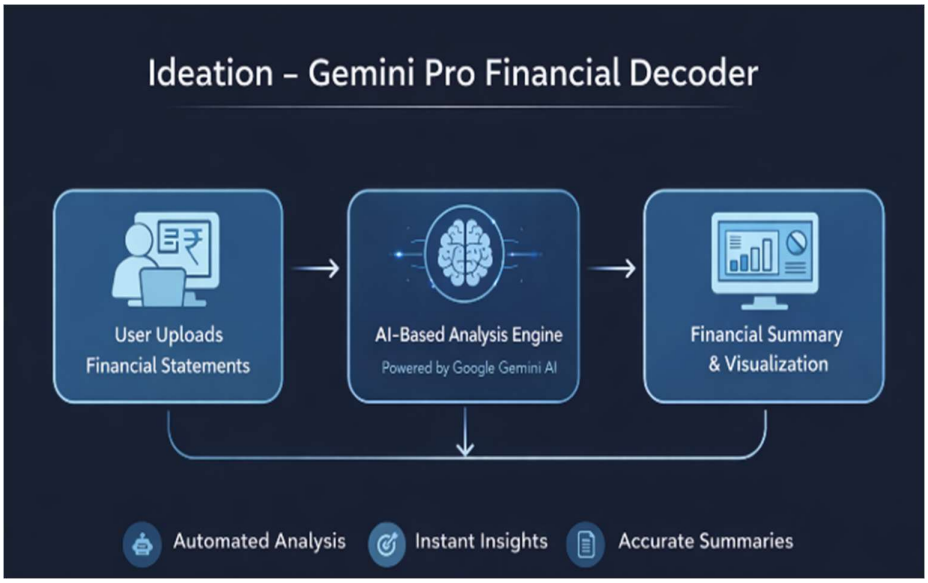
After evaluating feasibility, impact, and scalability, the AI-powered solution was selected.

Selected Idea

An AI-driven web application that analyzes financial statements and generates summaries and visual insights using Google Gemini AI.

Objectives of the Project

- To automate financial statement analysis
- To reduce dependency on financial experts
- To provide easy-to-understand summaries
- To visualize financial data for better decision-making



Scope of the Project

The scope of the **Gemini Pro Financial Decoder** project defines the boundaries and functionalities of the proposed system. The project focuses on analyzing financial statements using artificial intelligence and presenting insights in a simple and understandable format.

The scope of the project includes:

- Uploading financial statements such as Balance Sheet, Profit and Loss, and Cash Flow statements.
- Automated analysis of uploaded financial data.
- Generation of textual summaries explaining financial performance.
- Visualization of numerical data using charts and tables.
- Web-based implementation for easy access through a browser.

The project does not include advanced accounting audits, legal financial compliance checks, or real-time financial forecasting.

Stakeholders Identification

The stakeholders involved in the project are:

- **Students and Learners**: To understand financial concepts and statements easily.

- **Small Business Owners**: To analyse business financial health without expert assistance.
- **Project Evaluators**: To assess the functionality and implementation of the system.
- **Developers**: Responsible for design, development, testing, and maintenance of the application.

Assumptions

The project is developed based on the following assumptions:

- Uploaded files are in correct CSV or XLSX format.
- Financial data contains proper column names and numerical values.
- Users have internet connectivity to access AI services.
- The system is used for educational and analytical purposes only.

Constraints

The project has certain limitations:

- AI-generated summaries depend on API availability and quota limits.
- The system supports limited file formats.
- Accuracy depends on the quality of uploaded data.
- Complex financial interpretations may not be fully covered.

Expected Outcomes

The expected outcomes of the project are:

- Quick and automated financial analysis.
- Clear and meaningful financial summaries.
- Visual representation of financial data for better understanding.
- Reduced dependency on financial experts for basic analysis.
- Improved decision-making support for users.

The ideation phase successfully identified a real-world problem and proposed an AI-driven solution. The selected idea is feasible, impactful, and scalable. The project sets a strong foundation for further phases such as requirement analysis, system design, and development.

Define the Problem Statements

Customer Problem Statement – Introduction

In today's digital environment, individuals and small organizations generate and manage large volumes of financial data in the form of Balance Sheets, Profit and Loss Statements, and Cash Flow Statements. Although this data is critical for decision-making, interpreting financial statements accurately requires accounting knowledge and professional expertise.

Students, startup founders, small business owners, and non-finance professionals often struggle to understand financial documents due to their complex structure and technical terminology. Manual analysis is time-consuming, error-prone, and highly dependent on financial experts.

This project aims to identify these challenges from the customer's perspective and define clear problem statements that justify the need for an **AI-powered financial analysis system**

Customer Perspective Analysis

I am (Customer)	I'm trying to	But	Because	Which makes me feel
A student, startup founder, small business owner, or non-finance professional who needs to understand financial statements.	Analyze financial statements to understand business performance, profitability, and financial stability.	I Lack accounting knowledge and struggle to interpret complex financial terms and numerical data.	Traditional financial reports are technical lengthy, and require professional expertise for accurate analysis.	Confused, dependant on financial experts, and uncertain while making financial decisions.
But	Analyze financial statements to understand business performance, profitability, and financial stability.		Traditional financial reports are technical, lengthy, and require professional expertise for accurate analysis.	
Because	Confused, dependent on financial experts, and uncertain while making financial decisions.			

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A small business owner	Understand profit, expenses, and cash flow	I cannot easily analyze financial reports	Financial statements are complex and technical	Confused and dependent on accountants
PS-2	A student or beginner	Learn financial analysis practically	I struggle to interpret real financial data	Reports are not beginner-friendly	Overwhelmed and discouraged

Empathize & Discover

2.2Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

What does the user THINK & FEEL?

- Feels financial reports are difficult and confusing
- Worries about making wrong financial decisions
- Feels stressed due to lack of accounting knowledge
- Wants quick and simple understanding of business health

What does the user SEE?

- Large Excel sheets with numbers
- Complex balance sheets and profit & loss statements
- Financial dashboards that assume accounting knowledge
- Technical terms without explanations

What does the user HEAR?

- Advice from accountants and financial experts
- Suggestions to "consult a professional"
- Feedback that financial analysis is complex
- Opinions that finance is not easy for beginners

What does the user SAY & DO?

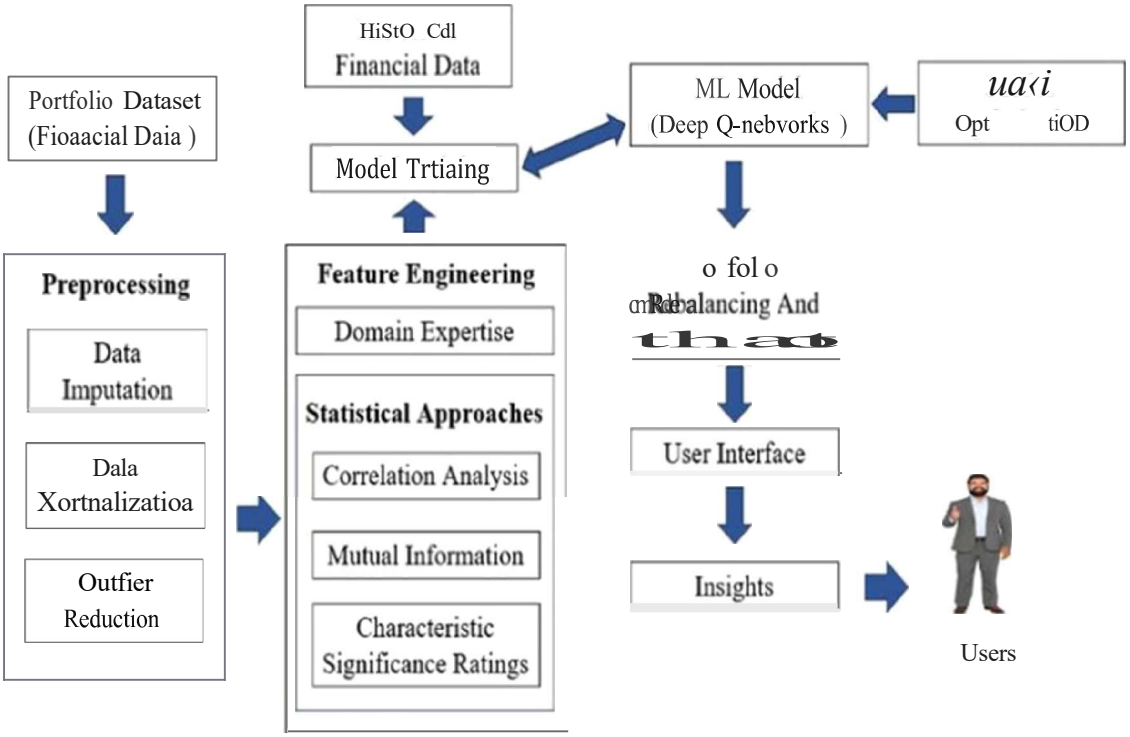
- Tries to analyze statements manually
- Searches online for financial explanations
- Depends on experts for clarity
- Uses basic tools like Excel without insights

▼ PAINS

- Difficulty understanding financial terminology
- Time-consuming manual analysis
- Dependency on financial experts
- Fear of incorrect financial decisions

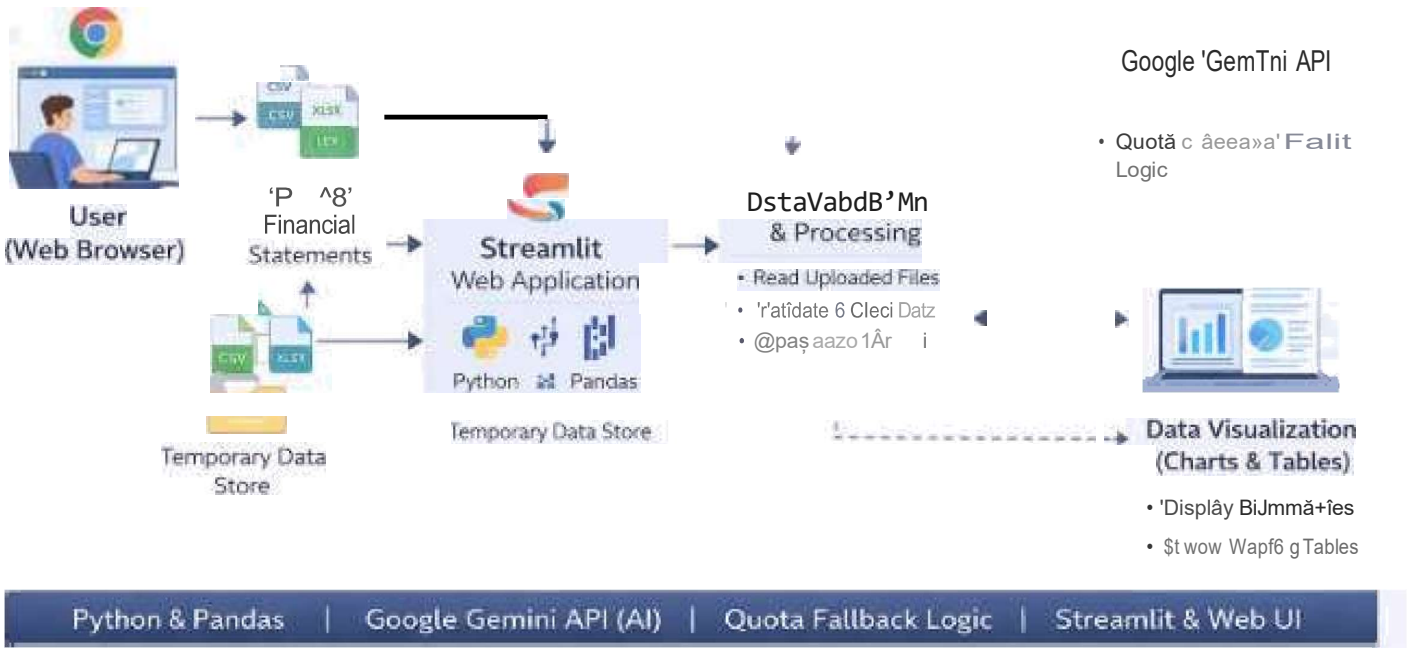
✸ GAINS

- Clear and simple financial summaries
- Faster decision-making
- Reduced dependency on experts
- Visual insights into financial data


HiStO Cdl
Financial Data
Portfolio Dataset
(Fioaacial Daia)
Model Trtiaing
ML Model
(Deep Q-nebvorks)
ua‹i
Opt tiOD
Preprocessing
Data Imputation
Dala Xortnalizatioa
Outfier Reduction
Feature Engineering
Domain Expertise
Statistical Approaches
Correlation Analysis
Mutual Information
Characteristic Significance Ratings
o fol o
Rbalancing And
thad
User Interface
Insights
Users


Key Operational Metrics
Interactive Charts, Tables and Data Elements
AI-powered Recommendations
Built For Mobile
Margin Analysis
Qlik
14,014
537,829
5.31%
767

Gemini Pro Financial Decoder System Architecture



Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Access to System	User can open the web application through a browser
FR-2	Upload Financial Statements	Upload Balance Sheet (CSV / XLSX) Upload Profit and Loss Statement (CSV / XLSX)
FR-3	Data Processing	Read and validate uploaded financial files Handle missing or invalid data safely
FR-4	Financial Analysis	Analyse Balance Sheet data Analyse Profit and Loss data Analyse Cash Flow data
FR-5	Summary Generation	Generate AI-based summary using Gemini API Generate local rule-based summary when AI quota is exceeded
FR-6	Data Visualization	Display financial data in tables Generate charts and graphs for numeric data
FR-7	Report Generation	Generate summaries and visuals on button click
FR-8	Error Handling	Display user-friendly error messages
FR-9	System Feedback	Show loading indicators during processing

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should be easy to use for non-finance users
NFR-2	**Security**	API keys must be protected and not exposed to users
NFR-3	**Reliability**	System should generate summaries even when AI quota fails
NFR-4	**Performance**	Summaries and charts should load within a few seconds
NFR-5	**Availability**	Application should be accessible whenever internet is available
NFR-6	**Scalability**	System should support larger financial datasets

Project Design Phase-II
Data Flow Diagram & User Stories

Data Flow Diagrams:

A **Data Flow Diagram (DFD)** represents how data flows within the *Gemini Pro Financial Decoder* system.
It shows how financial data is **input**, **processed**, **analyzed**, **stored**, and **presented** to the user.

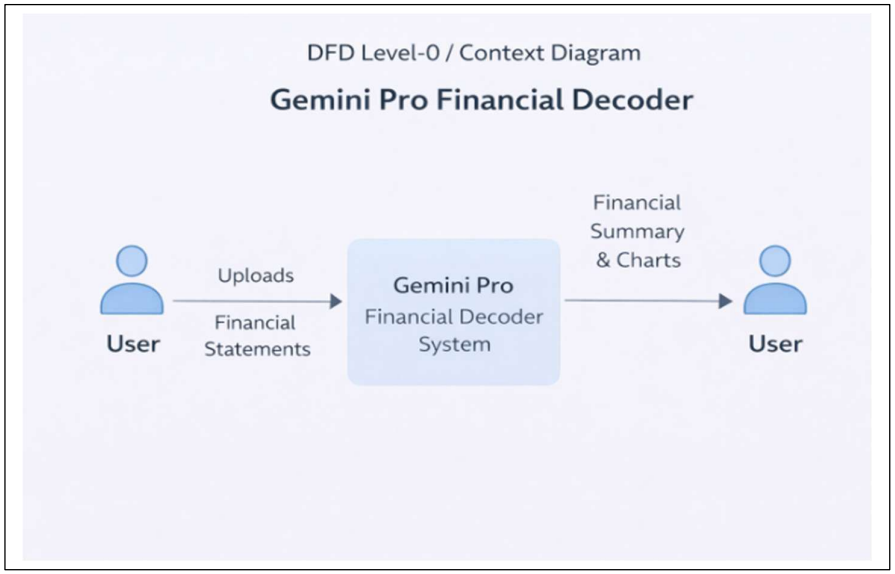
The system accepts financial statements such as **Balance Sheet**, **Profit & Loss**, and **Cash Flow** files and processes them using AI-based analysis and rule-based logic to generate **summaries and visual insights**.

Example: (Simplified)

Description:

- The User uploads financial statement files.
- The Gemini Pro Financial Decoder System processes the uploaded data.
- The system generates financial summaries and visualizations.
- The results are displayed back to the User.

Example: DFD Level 0



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Web User)	File Upload	USN-1	As a user, I want to upload Balance Sheet, Profit & Loss, and Cash Flow files so that I can analyze financial data.	Files are uploaded successfully in CSV/XLSX format	High	Sprint-1
Customer (Web User)	Data Processing	USN-2	As a user, I want the system to validate and read my uploaded files correctly.	System reads data without errors	High	Sprint-1
Customer (Web User)	Financial Analysis	USN-3	As a user, I want the system to analyze financial statements automatically.	Summary is generated correctly	High	Sprint-1
Customer (Web User)	AI Summary	USN-4	As a user, I want AI-based insights for better understanding.	AI or fallback summary displayed	Medium	Sprint-2
Customer (Web User)	Visualization	USN-5	As a user, I want charts and tables to visualize financial performance.	Graphs load correctly	Medium	Sprint-2
Administrator	System Monitoring	USN-6	As an admin, I want to ensure system stability and performance.	System runs without crashes	Low	Sprint-3

Project Design Phase-II
Technology Stack (Architecture & Stack)

Technical Architecture:

The Gemini Pro Financial Decoder follows a web-based, AI-driven architecture where users upload financial statements that are processed using application logic and analyzed using Google Gemini Generative AI. The system generates simplified summaries and visual insights to assist decision-making for non-finance users.

The architecture clearly separates:

- User Interface
- Application Logic
- External AI Services
- Data Processing & Visualization

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web interface where users upload financial statements and view summaries & charts	Streamlit (Python Web Framework), HTML, CSS
2	Application Logic-1	Handles file upload, validation, and preprocessing of CSV/XLSX files	Python
3	Application Logic-2	Processes financial data and prepares prompts for AI analysis	Python, Pandas

S.No	Component	Description	Technology
4	Application Logic-3	Controls workflow between UI, AI analysis, and visualization	Python
5	Database	Temporary in-memory data handling for uploaded financial files	Pandas DataFrames
6	Cloud Database	Not used; data processed locally for privacy	N/A
7	File Storage	Temporary storage of uploaded financial statements during runtime	Local File System
8	External API-1	Generates financial summaries using Generative AI	Google Gemini API
9	External API-2	Not applicable	N/A
10	Machine Learning Model	Natural Language Generation for financial summaries	Gemini Pro / Gemini Flash
11	Infrastructure (Server / Cloud)	Application runs locally or can be deployed on cloud	Local System / Google Cloud

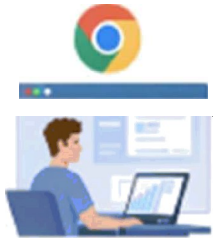
---

Table-2: Application Characteristics

S.No	Characteristic	Description	Technology Used
1	Open-Source Frameworks	Uses open-source tools for UI and data processing	Python, Streamlit, Pandas
2	Security Implementations	API key-based access control and secure data handling	Google API Key, HTTPS
3	Scalable Architecture	Modular design allows easy scaling of AI and UI components	Python Modular Architecture
4	Availability	Application is available as long as local server or cloud instance is running	Localhost / Cloud VM
5	Performance	Efficient data handling with limited rows sent to AI; visualizations rendered locally	Pandas, Streamlit

▯

Gemini Pro Financial Decoder Architecture



User
(Browser)

Uptoads Financial Statements

Streamlit
Web Application

- Python-based interface

Fi(c upl¢›adir›g & UI

HTML, CSS, JavaSeript / Angular Js / React 1s eic.

Python-based interface

File utxoadi• g & Ui

Initial data processing

Application Logic
(Python + Pandas)

Sends Prompts

- Fite validation
- Data analysis
- Prompt generation
- Python

Goog(e
Gem"ini API

- Generative AI Engine
- Text-bgy@ finanCial analyis
- Google Gemini API

A’*! "*

Visualization Output
(Charts & Summaries)

- Financial dJta v›suul zutien
- S›mpl fied financial
- Loc al Syste•m / Googtc VM

- Generative AI Engine
- Aadiwr API. etc.
- Local, Server Configuration

AI-Driven Analysis | Secure Data Processing | Web-Based Application | Results for Non-Finance Users

Project Design Phase
Problem – Solution Fit

Date	15 February 2025
Team ID	LTVIP2026TMIDS36980
Project Name	Google Pro Financial Decoder
Maximum Marks	2 Marks

Problem – Solution Fit:

This Problem–Solution Fit template explains the core user problems in financial statement analysis and how the *Gemini Pro Financial Decoder* addresses them using AI-based automation and intelligent visualizations.

Purpose:

- To automate the analysis of financial statements using technology
- To simplify complex financial data for non-finance users
- To reduce dependency on accountants and financial experts
- To provide quick and accurate financial summaries
- To help users understand profit, loss, and cash flow easily
- To visualize financial data for better clarity and insights
- To support better and faster business decision-making
- To make financial analysis accessible to students and small businesses

Template:

1 **Problem Statement** (Problem to be solved)	Financial statements such as Balance Sheet, Profit & Loss, and Cash Flow are complex and difficult to interpret for students, small business owners, and non-finance professionals, leading to confusion and poor decision-making.
2 **Idea / Solution Description**	To develop a web-based AI application called **Gemini Pro Financial Decoder** that allows users to upload financial statements and automatically generates summaries and visualizations using AI and rule-based logic.
3 **Novelty / Uniqueness**	The application combines Generative AI with a local fallback summary system, ensuring uninterrupted insights even when AI quota limits are exceeded, which makes it reliable
4 **Social Impact / Customer Satisfaction**	The solution empowers students, startups, and small businesses to independently understand financial data, reducing dependency on financial experts and improving confidence in decision-maing.
5 **Business Model** (Revenue Model)	The system can be offered as a freemium platform with basic analysis for free and advanced analytics, exports, and enterprise features available through subscription plans.
6 **Scalability of the Solution**	The application is highly scalable as it is built using cloud-ready technologies and can support more users, datasets, and additional financial modules with minimal changes.

Project Design Phase
Proposed Solution

Proposed Solution :

The following table describes the proposed solution for automating financial statement analysis using Artificial Intelligence to provide simple summaries and visual insights for non-finance users.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Financial statements such as Balance Sheets, Profit and Loss Statements, and Cash Flow Statements are complex and difficult to understand for non-finance users, leading to poor financial decision-making.
2.	Idea / Solution description	To develop a web-based AI application that allows users to upload financial statements and automatically generates easy-to-understand summaries and visualizations using Google Gemini AI with a fallback rule-based system.
3.	Novelty / Uniqueness	The solution combines AI-powered financial analysis with a local fallback mechanism, ensuring continuous summary generation even when AI quota limitations occur.
4.	Social Impact / Customer Satisfaction	Helps students, startups, and small business owners understand financial data independently, reducing dependency on financial experts and improving confidence in decision-making.
5.	Business Model (Revenue Model)	The application can follow a freemium model with basic features for free and advanced analytics, report exports, and premium insights available through paid subscriptions.
6.	Scalability of the Solution	The solution is scalable as it is web-based and can support a large number of users by deploying on cloud infrastructure with optimized data processing and AI integration.

Project Design Phase
Solution Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

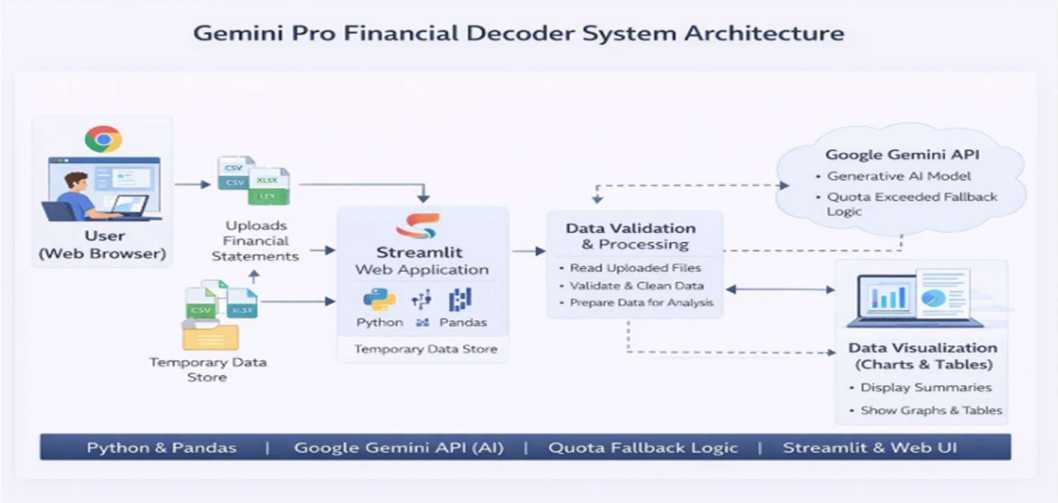


Figure 1: Architecture and data flow of the Google Pro Financial Decoder

The project planning phase defines how the **Gemini Pro Financial Decoder** will be executed using an **Agile–Scrum methodology**.
The planning focuses on sprint-wise task allocation, story point estimation, team velocity, and delivery tracking to ensure timely completion of the project.

The project is divided into **multiple sprints**, each with clearly defined goals, tasks, and measurable outcomes.

---

2. Agile Planning Approach

The project follows **Scrum-based Agile planning** with the following characteristics:

- Sprint-based development
- Story point estimation
- Velocity tracking
- Incremental feature delivery
- Continuous testing and improvement

---

3. Sprint Configuration

Parameter	Value
Sprint Duration	1 Week
Number of Sprints	4
Team Size	1–2 Members
Working Days per Sprint	5
Daily Working Hours	4 Hours

---

4. Velocity Planning

Velocity Definition

Velocity is the number of **story points completed per sprint**.

Based on skill level, project complexity, and academic constraints:

Sprint	Planned Velocity (Story Points)
Sprint 1	12
Sprint 2	15
Sprint 3	18
Sprint 4	15

Average Velocity = 15 Story Points / Sprint

---

5. Product Backlog (High-Level)

Backlog ID	Feature	Priority
PB-1	Financial file upload	High
PB-2	CSV/XLSX data processing	High
PB-3	Gemini AI summary generation	High
PB-4	Fallback summary logic	Medium
PB-5	Data visualization	Medium
PB-6	UI enhancements	Low
PB-7	Testing and optimization	High

---

6. Sprint-wise Planning Table (IMPORTANT)

🔹 Sprint 1 – Requirement & UI Setup

User Story ID	Task Description	Story Points
US-1	Requirement analysis and scope definition	3
US-2	UI layout using Streamlit	4
US-3	File upload components	5
Total		**12 SP**

🔹 Sprint 2 – Data Processing & AI Integration

User Story ID	Task Description	Story Points
US-4	CSV/XLSX parsing using Pandas	5
US-5	Gemini API integration	5
US-6	Model initialization & testing	5
Total		**15 SP**

🔹 Sprint 3 – Summary Logic & Visualization

User Story ID	Task Description	Story Points
US-7	AI-based summary generation	6
US-8	Local rule-based fallback summary	4
US-9	Data visualization charts	5
US-10	UI refinement	3
Total		**18 SP**

🔹 Sprint 4 – Testing, Optimization & Deployment

User Story ID	Task Description	Story Points
US-11	Error handling & validation	4
US-12	API quota handling	4
US-13	Performance testing	4
US-14	Final deployment & demo	3
Total		**15 SP**

---

7. Sprint Capacity Planning

Sprint	Available Hours	Story Points Planned
Sprint 1	20 hrs	12
Sprint 2	20 hrs	15
Sprint 3	20 hrs	18
Sprint 4	20 hrs	15

---

8. Burndown Logic (Textual Explanation)

During each sprint:

- Tasks are tracked daily
- Completed story points reduce remaining workload
- Velocity is used to predict next sprint capacity

This ensures:

- No overload
- Predictable delivery
- Controlled progress

---

9. Risk-Based Planning Logic

Risk	Impact	Planned Control
Gemini API quota exhaustion	Summary failure	Local fallback logic
Large datasets	Performance drop	Data sampling
Invalid file format	App crash	File validation
API latency	Delay	Retry handling

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Product Backlog, Sprint Schedule, and Estimation

Sprint	Epic	User Story ID	User Story Description	Story Points	Priority
Sprint 1	File Upload	US-1	As a user, I can upload balance sheet, P&L, and cash flow files	3	High
Sprint 1	Data Processing	US-2	As a user, I want the system to read and validate uploaded files	2	High
Sprint 2	AI Summary	US-3	As a user, I want AI-generated summaries for financial data	5	High
Sprint 2	Fallback Logic	US-4	As a user, I want summaries even when AI quota is exceeded	3	Medium
Sprint 3	Visualization	US-5	As a user, I want charts for financial insights	3	High
Sprint 3	UI	US-6	As a user, I want a simple and clean web interface	2	Medium

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Status
Sprint 1	5	7 Days	Completed
Sprint 2	8	7 Days	Completed
Sprint 3	5	7 Days	Completed

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Functional & Performance Testing Template

Model Performance Test

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	File Upload Validation (CSV/XLSX)	Upload valid and invalid file formats in Balance Sheet, P&L, and Cash Flow sections	Only CSV/XLSX files are accepted	Valid formats accepted, others rejected	Pass
FT-02	Financial Data Reading	Upload files with financial data and click “Generate Reports”	System should correctly read and display data	Data displayed correctly in tables	Pass
FT-03	Summary Generation (AI / Fallback)	Upload files and generate summary	Summary should be generated using Gemini or local logic	Summary generated successfully using fallback logic	Pass
FT-04	Visualization Generation	Check if API key is correct and model responds	Graphs should be created from numeric data	Line charts generated correctly	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	No performance degradation	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	No performance degradation	Pass

PT-03	Concurrent Inputs	Upload multiple files repeatedly	App should handle repeated usage	No performance degradation	Pass

Acceptance Testing
UAT Execution & Report Submission

1. Purpose of Document

This table represents the phase-wise project planning, activities, duration, and deliverables for the successful development of the Gemini Pro Financial Decoder.

User Acceptance Testing (UAT) is performed to validate whether the Gemini Pro Financial Decoder meets user requirements and works correctly from an end-user perspective.

Test Case ID	Test Scenario	User Action	Expected Result	Actual Result	Status
UAT-01	Upload Balance Sheet file	User uploads a valid CSV/XLSX balance sheet file	File should upload successfully	File uploaded successfully	Pass
UAT-02	Upload Profit & Loss file	User uploads profit and loss statement	File should be accepted without errors	File uploaded successfully	Pass
UAT-03	Upload Cash Flow file	User uploads cash flow statement	File should upload and be readable	File uploaded successfully	Pass
UAT-04	Invalid file format	User uploads unsupported file (PDF/TXT)	System should reject file	File rejected with message	Pass
UAT-05	Generate Reports button	User clicks **Generate Reports**	System should process data	Reports generated	Pass
UAT-06	AI summary generation	User generates summaries with Gemini available	AI summary should appear	AI summary displayed	Pass
UAT-07	API quota exceeded	Gemini quota exhausted	System should fallback to local summary	Local summary generated	Pass
UAT-08	Data visualization	User views charts	Charts should display correctly	Charts displayed	Pass
UAT-09	Large dataset handling	User uploads large CSV file	App should not crash	Data handled successfully	Pass
UAT-10	Overall application flow	User performs complete workflow	Smooth navigation and output	Application works as expected	Pass

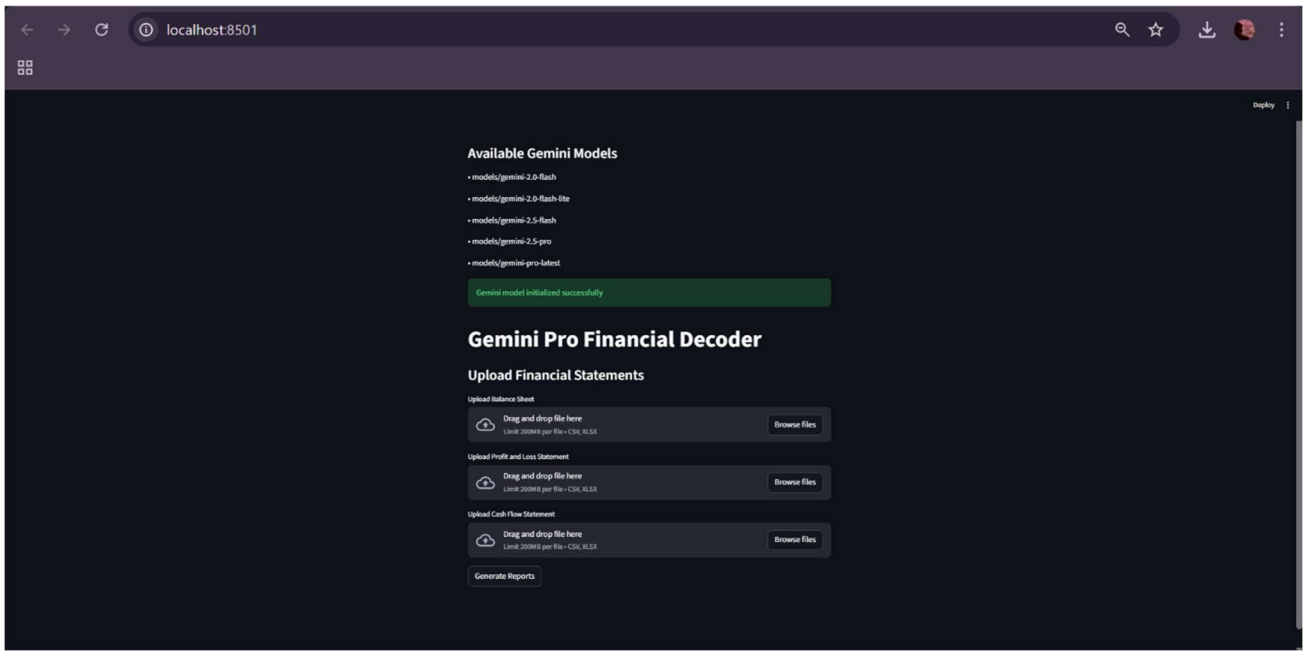
UAT Summary Table

Parameter	Result
Total Test Cases Executed	10
Test Cases Passed	10
Test Cases Failed	0
Acceptance Status	**Accepted**

Based on the User Acceptance Testing results, the Gemini Pro Financial Decoder meets all functional and usability requirements. The system performs reliably under normal and exceptional conditions, including API quota limitations. Hence, the application is approved for final deployment.

7.RESULTS

7.1 Output Results



Deploy

Balance Sheet Summary

Total Balance Value: ₹3,860,000

The balance sheet shows overall financial position. Assets and liabilities appear structured.

Balance Sheet Data

	Category	Item	Amount
0	Assets	Cash	120000
1	Assets	Bank Balance	350000
2	Assets	Accounts Receivable	230000
3	Assets	Inventory	180000
4	Assets	Prepaid Expenses	40000
5	Assets	Short Term Investments	90000
6	Assets	Plant and Machinery	750000
7	Assets	Office Equipment	130000
8	Assets	Furniture	85000
9	Assets	Intangible Assets	95000

Deploy

Profit and Loss Summary

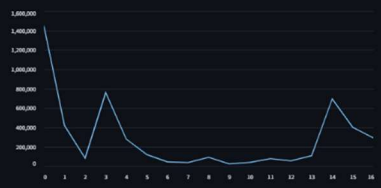
Total Revenue: ₹1,955,000 Total Expenses: ₹1,534,000

Net Profit: ₹421,000

The company shows good profitability.

Profit and Loss Data

	Category	Item	+ Amount
9	Expenses	Office Supplies	25000
7	Expenses	Internet and IT	38000
10	Expenses	Maintenance	40000
6	Expenses	Electricity	45000
12	Expenses	Interest Expense	56000
11	Expenses	Depreciation	78000
2	Revenue	Other Income	85000
8	Expenses	Marketing	92000
13	Taxes	Income Tax	110000
5	Expenses	Rent	120000



Deploy

Cash Flow Summary

Total Cash Movement: ₹90,000

Cash flow shows liquidity movement and operations.

Cash Flow Data

	Category	Activity	Amount
0	Operating	Net Profit	292000
1	Operating	Depreciation	78000
2	Operating	Change in Inventory	-65000
3	Operating	Change in Receivables	-72000
4	Operating	Change in Payables	48000
5	Operating	Interest Paid	-56000
6	Operating	Tax Paid	-110000
7	Investing	Purchase of Equipment	-220000
8	Investing	Sale of Assets	45000
9	Investing	Investments Made	-95000

8. ADVANTAGES & DISADVANTAGES

Advantages

- Easy to use for non-finance users
- Automated financial analysis
- Reliable fallback mechanism
- Clear visual insights
- Saves time and effort

Disadvantages

- AI summary depends on API quota
- Limited to structured CSV/XLSX data

---

9. CONCLUSION

The Gemini Pro Financial Decoder successfully simplifies financial statement analysis using Generative AI and intelligent fallback logic. for real-world financial analysis scenarios.

---

10. FUTURE SCOPE

- PDF report export
- Multi-language support
- Advanced financial ratio analysis
- Cloud deployment
- Role-based user authentication

---

11. APPENDIX

Source code:

```
import streamlit as st
import pandas as pd
import google.generativeai as genai

# ================================
# CONFIGURATION
# ================================
```

```
GOOGLE_API_KEY = "AIzaSyBoGX0hGHC5-QQrHbqjdyLmETUFYXeLX0Y"
genai.configure(api_key=GOOGLE_API_KEY)

# Fixed model list (safe – no API call)
AVAILABLE_MODELS = [
    "models/gemini-2.0-flash",
    "models/gemini-2.0-flash-lite",
    "models/gemini-2.5-flash",
    "models/gemini-2.5-pro",
    "models/gemini-pro-latest"
]

# Try initializing Gemini model
try:
    model = genai.GenerativeModel("models/gemini-2.0-flash")
    gemini_available = True
except Exception:
    model = None
    gemini_available = False

# ================================
# UI – MODEL LIST
# ================================

st.subheader("Available Gemini Models")
for m in AVAILABLE_MODELS:
    st.write("•", m)

if gemini_available:
    st.success("Gemini model initialized successfully")
```

```
else:
    st.warning("Gemini quota unavailable. Using local summary.")

# ================================
# PROMPT TEMPLATES
# ================================

prompt_templates = {
    "balance_sheet": """
You are a financial analyst.
Given the balance sheet data below:
{data}

Explain financial position, assets vs liabilities, and risks.
""",

    "profit_loss": """
You are a financial analyst.
Given the profit and loss data below:
{data}

Explain revenue trend, expenses, and profitability.
""",

    "cash_flow": """
You are a financial analyst.
Given the cash flow data below:
{data}

Explain operating, investing, and financing cash flow.
"""
```

```
}

# ================================
# LOCAL FALLBACK SUMMARY (NO API)
# ================================

def local_financial_summary(df, statement_type):
    if df is None:
        return "No data uploaded."

    if "Amount" not in df.columns:
        return "Invalid file format."

    total = df["Amount"].sum()

    if statement_type == "profit_loss":
        revenue = df[df["Category"] == "Revenue"]["Amount"].sum()
        expenses = df[df["Category"] == "Expenses"]["Amount"].sum()
        profit = revenue - expenses

        return f"""
Total Revenue: ₹{revenue:,}
Total Expenses: ₹{expenses:,}

Net {'Profit' if profit > 0 else 'Loss'}: ₹{profit:,}

The company shows {'good profitability' if profit > 0 else 'financial stress'}.
"""

    if statement_type == "balance_sheet":
        return f"""
```

```
Total Balance Value: ₹{total:,}

The balance sheet shows overall financial position.
Assets and liabilities appear structured.
"""

    if statement_type == "cash_flow":
        return f"""
Total Cash Movement: ₹{total:,}

Cash flow shows liquidity movement and operations.
"""

    return "Summary generated."

# ================================
# FILE HANDLING
# ================================

def load_file(file):
    if file is None:
        return None
    if file.name.endswith(".csv"):
        return pd.read_csv(file)
    if file.name.endswith(".xlsx"):
        return pd.read_excel(file)
    return None

# ================================
# SUMMARY GENERATION (SAFE)
# ================================
```

```
def generate_summary(statement_type, df):
    if df is None:
        return "No data uploaded."

    if not gemini_available:
        return local_financial_summary(df, statement_type)

    try:
        prompt = prompt_templates[statement_type].format(
            data=df.head(20).to_dict()
        )
        response = model.generate_content(prompt)
        return response.text
    except Exception:
        return local_financial_summary(df, statement_type)

# ================================
# VISUALIZATION
# ================================

def create_visuals(df, title):
    if df is not None:
        st.subheader(title)
        st.write(df)
        numeric_df = df.select_dtypes(include=["number"])
        if not numeric_df.empty:
            st.line_chart(numeric_df)

# ================================
# STREAMLIT APP UI
```

```
# ================================

st.title("Gemini Pro Financial Decoder")

st.subheader("Upload Financial Statements")

bs_file = st.file_uploader("Upload Balance Sheet", type=["csv", "xlsx"])
pl_file = st.file_uploader("Upload Profit and Loss Statement", type=["csv", "xlsx"])
cf_file = st.file_uploader("Upload Cash Flow Statement", type=["csv", "xlsx"])

if st.button("Generate Reports"):
    with st.spinner("Generating summaries and visualizations..."):

        bs_data = load_file(bs_file)
        pl_data = load_file(pl_file)
        cf_data = load_file(cf_file)

        bs_summary = generate_summary("balance_sheet", bs_data)
        pl_summary = generate_summary("profit_loss", pl_data)
        cf_summary = generate_summary("cash_flow", cf_data)

        st.subheader("Balance Sheet Summary")
        st.write(bs_summary)
        create_visuals(bs_data, "Balance Sheet Data")

        st.subheader("Profit and Loss Summary")
        st.write(pl_summary)
        create_visuals(pl_data, "Profit and Loss Data")

        st.subheader("Cash Flow Summary")
        st.write(cf_summary)
```

create_visuals(cf_data, "Cash Flow Data")

Datsets:

	A	B	C	D	E
1	Category	Item	Amount		
2	Assets	Cash	120000		
3	Assets	Bank Balance	350000		
4	Assets	Accounts Receivable	210000		
5	Assets	Inventory	180000		
6	Assets	Prepaid Expenses	40000		
7	Assets	Short Term Investments	90000		
8	Assets	Plant and Machinery	750000		
9	Assets	Office Equipment	130000		
10	Assets	Furniture	85000		
11	Assets	Intangible Assets	95000		
12	Liabilities	Accounts Payable	160000		
13	Liabilities	Short Term Loans	120000		
14	Liabilities	Accrued Expenses	70000		
15	Liabilities	Long Term Debt	450000		
16	Liabilities	Deferred Tax	60000		
17	Equity	Share Capital	500000		
18	Equity	Retained Earnings	315000		
19	Equity	Reserves	135000		
20					

	A	B	C	D	E
1	Category	Item	Amount		
2	Revenue	Product Sales	1450000		
3	Revenue	Service Income	420000		
4	Revenue	Other Income	85000		
5	Expenses	Cost of Goods Sold	760000		
6	Expenses	Salaries	280000		
7	Expenses	Rent	120000		
8	Expenses	Electricity	45000		
9	Expenses	Internet and IT	38000		
10	Expenses	Marketing	92000		
11	Expenses	Office Supplies	25000		
12	Expenses	Maintenance	40000		
13	Expenses	Depreciation	78000		
14	Expenses	Interest Expense	56000		
15	Taxes	Income Tax	110000		
16	Profit	Gross Profit	695000		
17	Profit	Operating Profit	402000		
18	Profit	Net Profit	292000		
19					
20					
21					

	A	B	C	D
1	Category	Activity	Amount	
2	Operating	Net Profit	292000	
3	Operating	Depreciation	78000	
4	Operating	Change in Inventory	-65000	
5	Operating	Change in Receivables	-72000	
6	Operating	Change in Payables	48000	
7	Operating	Interest Paid	-56000	
8	Operating	Tax Paid	-110000	
9	Investing	Purchase of Equipment	-220000	
10	Investing	Sale of Assets	45000	
11	Investing	Investments Made	-95000	
12	Financing	Loan Received	180000	
13	Financing	Loan Repaid	-90000	
14	Financing	Equity Issued	150000	
15	Financing	Dividends Paid	-70000	
16	Summary	Net Cash Flow	75000	
17				
18				

Project Demo Video Link:

https://drive.google.com/file/d/15JIGUu2NSbxOQes8kUzq5GLP6ykM3_95/view?usp=sharing