# PROJECT REPORT

## 1.COVER PAGE:

PROGRAMM TITLE :Agriculture loan Eligibility checker.

COURSE NAME : Introduction to problem solving and programming.

SUBMITTED BY: (BUDDANNAGARI SHANMUGAM).

REGISTRATION NUMBER: 25MIB10018

FACULTY: (Dr .K. SIVABALAN).

INSTITUTION:(VIT BHOPAL).

SEMISTER :Fall semester.

SUBMISSION DATE: 23/11/2025

## 2.INTRODUCTION:

**This project (AGRICULTURE LOAN ELIGIBILITY CHECKER),is designed to help farmers, banking staff, and self help groups .whether a farmer qualifies for a loan based on the given criteria such as land area, annual income, and existing loan. The system simplifies the process. Agriculture remains the backbone of India's economy, farmers often depend upon the loans for irrigation, fertilizers, seeds, equipment, and other useful needs.**

## 3.PROBLEM STATEMENT:

Financial institutions offering agriculture loans need to verify the capacity of the farmer before giving the loans. we want to give it according to his financial situation.

The verification for checking the conditions is satisfy -

*Purpose to taking the loan.

*Annual farmer income.

*Previous loan dues.

*Land ownership and land area.

*Credit history and repayment behaviour of the farmer.

*Crop type and seasonal risk.

Many farmers also lack clarity about the eligibility requirements before applying.

HERE PROBLEM IS:

How we can develop an automated, reliable, and easy to use system that determines a farmers' eligibility for agricultural loans based on multiple criteria?

## 4.FUNCTIONAL REQUIREMENTS:

The functionalities are:

1.INPUT USER SECTION:

~Farmer details like name and age.

~Purpose of taking loan .

~Annual income in Indian rupees.

~Land area in acres.

~Type of farming like crop farming, dairy, poultry, etc.

~Existing loan amount.

~Credit score.

~Repayment behaviour.

~Loan amount requested.

2.VALIDATION FUNCTION:

~Prevent negative and unrealistic inputs.

~Check for missing or validation values.

3.ELIGIBILITY EVALUTION:

~Check land area size and land owner ship.

~Check minimum bank account threshold.

~Analyse farmer's income .

~Compare existing loan burden .

~Determine suitable loan category.

4.DISPLAY THE RESULT:

~Eligible.

~Not eligible .

~Conditional eligibility.

~Display reasons for rejections .

5.DATA STORAGE:

~Save entered data into a file.

6.LOAN RECOMMENDATION:

~Provide improvement tips .

~Suggest some best loan schemes based on user inputs.

## 5.NON-FUNCTIONAL REQUIREMENTS:

PERFORMANCE:

~System must provide results within 2 seconds.

RELIABILITY :

~Decision making must be consistent and rule based.

MAINTAINABILITY:

~Code should be modular and easy to update with new policies.

SECURITY:

~Sensitive information like income and credit score must be protected.

~No data should be shared without permission.

USABILITY:

~Farmers and staff with minimum digital knowledge must be able to use it .

~Interface must be simple.

SCALABILITY:~Should support more schemes ,more loan types and advanced decision making models in the future.

## 6.SYSTEM ARCHITECTURE:

1.INPUT LAYER:

Collects farmer data through a form or console.

2.BUSINESS LOGIC LAYER:

Contains :

~Input validation module .

~Eligibility rule engine.

~Decision making algorithms.

Rules :

~Land area >=1 acre

~Annual income >= 60000

~Credit score >=750

~Existing loan <=200000

~Repayment capacity >= 30%

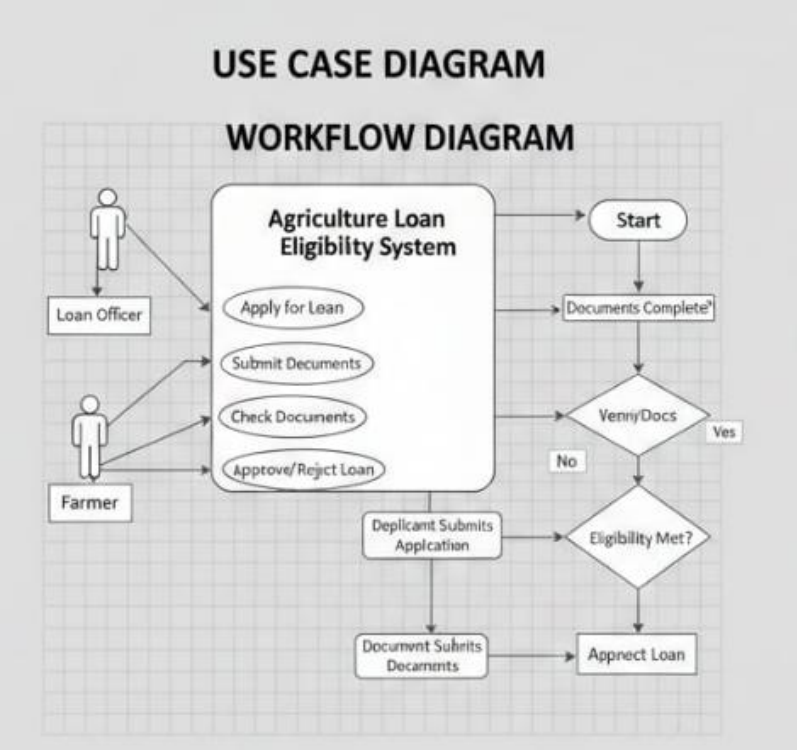
3.OUTPUT LAYER:

~Display eligibility result .

~Recommends loan categories .

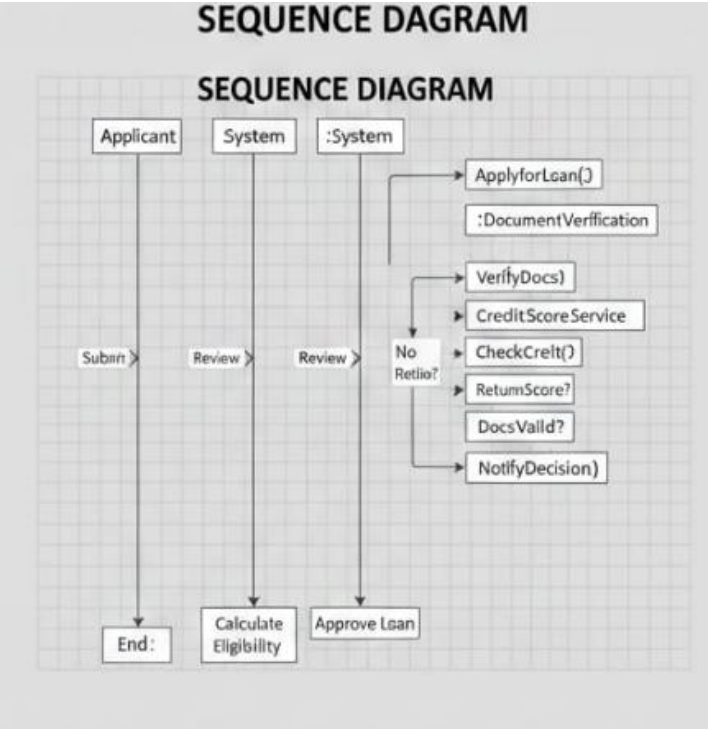
~Suggests improvement areas.

*Architecture ensures clarity, modularity, and reusability.

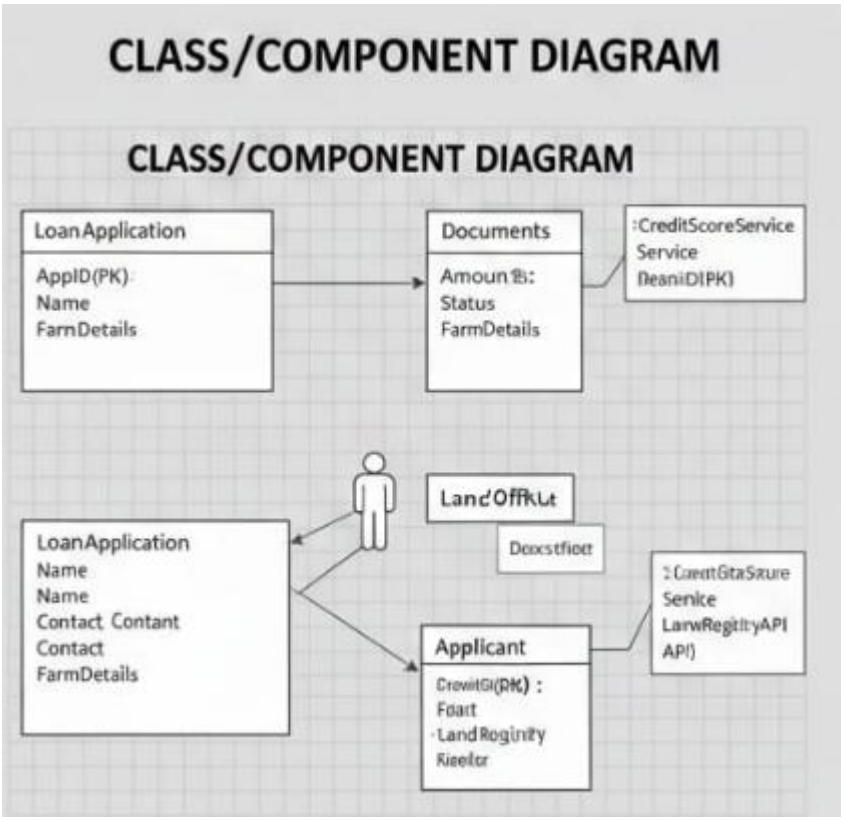



1.USE CASE DIAGRAM.

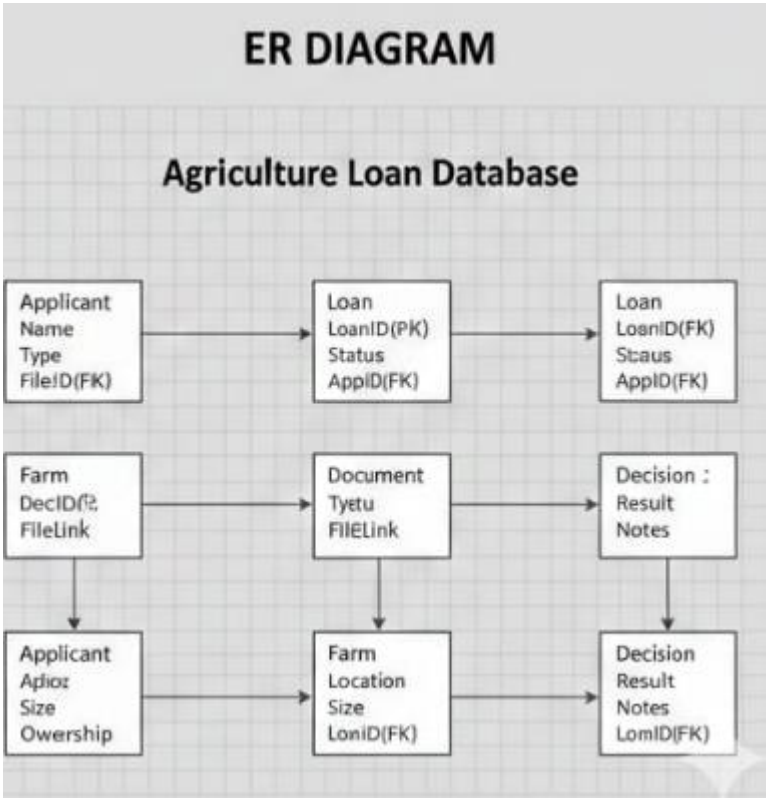
2.WORK FLOW DIAGRAM.




3.SEQUENCE DIAGRAM.




4.CLASS/COMPONENT DIAGRAM.




5.ER DIAGRAM.

## 8.DESIGN DECISIONS & RATIONALE

~File storage chosen to light weight nature and no setup needed.

~Python chosen due to readability and strong support libraries .

~Modular programming chosen to separate logic, input, and output .

~Rule based engine allows simple extension of policies.

~Chosen threshold based eligibility rules instead of ML to keep system simple and transparent.

These decisions make the system robust, easily maintainable and scalable.

## 9.IMPLEMENTATION DETAILS:

KEY MODULES:

*Farmer_input.py – handles user input.

*Validation .py -verifies data integrity.

*Eligibility _checker.py-core decision engine.

*Recommendation .py -suggests improvements.

*Storage.py – saves farmer improvements .

IMPORTANT ALGARITHMS:

*Conditional rule checking .

*Credit score analysis .

TECHNOLOGIES:

*Python3

*JSON/CSV

*flowchart and UML tools (for diagrams)

## 10.SCREENSHOTS/RESULTS:

```
AGRICULTURE LOAN ELIGIBILITY CHECKER
Enter Farmer's Name:  shannu
Enter the age of Farmer:  50
Enter the annual income of the Farmer in Indian Rupees:  70000
Enter the total land of the farmer in acres:  4
Enter the crop type from farmer:  peas
Please check loan eligibility:
Do you have any existing loan? (yes/no):  no
Checking your loan eligibility..... Please wait a second
-- Eligibility Result --
Farmer: shannu
****ELIGIBILITY STATUS: LOAN APPROVED!****
Message has sent to your mobile phone
Thank you visit again

AGRICULTURE LOAN ELIGIBILITY CHECKER
Enter Farmer's Name:  sridhar
Enter the age of Farmer:  19
Enter the annual income of the Farmer in Indian Rupees:  120000
Enter the total land of the farmer in acres:  2.5
Enter the crop type from farmer:  paddy
Please check loan eligibility:
Do you have any existing loan? (yes/no):  yes
Checking your loan eligibility..... Please wait a second
-- Eligibility Result --
Farmer: sridhar
**ELIGIBILITY STATUS: LOAN DENIED.** Reason: Existing loan detected. Please clear previous dues.
Recommendation: Please review the requirements or consult with a loan officer.

AGRICULTURE LOAN ELIGIBILITY CHECKER
Enter Farmer's Name:  indrajith sen
Enter the age of Farmer:  18
Enter the annual income of the Farmer in Indian Rupees:  3000000000000000
Enter the total land of the farmer in acres:  0
Enter the crop type from farmer:  sunflower
Please check loan eligibility:
Do you have any existing loan? (yes/no):  no
Checking your loan eligibility..... Please wait a second
-- Eligibility Result --
Farmer: indrajith sen
**ELIGIBILITY STATUS: LOAN DENIED.** Reason: At least 1 acre of land is required.
Recommendation: Please review the requirements or consult with a loan officer.
```

## 11.TESTING APPROACH:

This project used to test black-box testing, unit testing, and boundary value analysis.

BLACK BOX TESTING:

Tested system behaviour without inspecting code.

BOUNDARY TESTING:

Ex:-

Land = 1 acre

Income = 60000

Credit score = 750

UNIT TESTING:

Each function tested individually

INTEGRATION TESTING:

Input – validation—eligibility calculation-output

STRESS TESTING:

Checking system values

*Very high income.

*Multiple negative values.

## 12.CHALLENGES FACED:

*Ensuring input validation for all cases .

*Keeping codes modular yet simple.

*Time management during development.

*Designing UML diagrams.

*Choosing universal eligibility rules.

*Handling diverse farming conditions and income ranges.

## 13.LEARNING &KEY TAKEAWAYS:

*Knowledge about agricultural finance policies.

*Implementing scalable and rule based software solutions.

*Translating real world conditions into Algorithms .

*Understanding system architecture and UML diagramming.

*Improved python coding  and debugging skills.

*Understanding loan evaluation mechanisms.

## 14.FUTURE ENHANCEMENTS:

*Add AI/ML model for risk prediction.

*Add graphical dashboard.

*Add Multiplingual audio support for rural users

*Convert to mobile app using( flask/ Kivi)

*Integrate with government databases (like PM Kisan, land records).

*Include interest rate calculation module.

## 15.REFERCES:

*Government of India agriculture loan guidelines.

*Reserve bank of India priority sector lending norms.

*NABARD loan schemes.

*Classroom notes and lecture slides.