1. INTRODUCTION

1.1 Project Overview

HepaCare is an AI-powered web application designed to classify various types of liver grains using deep learning techniques, particularly Transfer Learning. The project leverages a pre-trained Convolutional Neural Network (CNN) model to accurately identify patient clinical
varieties from images uploaded by the user.

This solution bridges the gap between medical practices and modern AI by offering an intuitive platform that automates liver variety classification, replacing traditional manual methods that are often time-consuming and error-prone. The system is implemented using Python, TensorFlow/Keras for the backend model, and Flask for the web interface, providing an end-to-end pipeline from image input to class prediction.

1.2 Purpose

The purpose of HepaCare is to:

- Provide an accessible and intelligent platform for liver cirrhosis prediction that benefits patients, distributors, exporters, food laboratories, and quality control units.
- Minimize manual effort and errors in the grain identification process through automation.
- Enhance decision-making for liver sorting, packaging, and distribution based on liver condition.
- Reduce dependency on expensive lab analysis by introducing a low-cost, AI-based tool.
- Encourage digital transformation in healthcare, particularly in quality inspection and post-harvest processing.

By addressing the practical challenges in patient clinical identification, this application contributes to both efficiency and accuracy, ultimately supporting the larger goal of precision
healthcare.

2. Ideation Phase

2.1 Define the Problem Statements

Date	30 june 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Reference: https://miro.com/templates/customer-problem-statement/

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A rice farmer from a rural village	identify the type of ricesee ds I have befor e cultivation	I don't have access to lab testing or expert identification	it's expensive and not locally available	confused, uncertain, and worried about crop planning
PS-2	An agricultural extension officer or researcher	quickly identify and classify different rice types in the field	manual classification is time-consuming and not always accurate	grain types look visually similar to the naked eye	frustrated and slows down data collection and analysis

2. Ideation Phase

2.2 Empathize & Discover

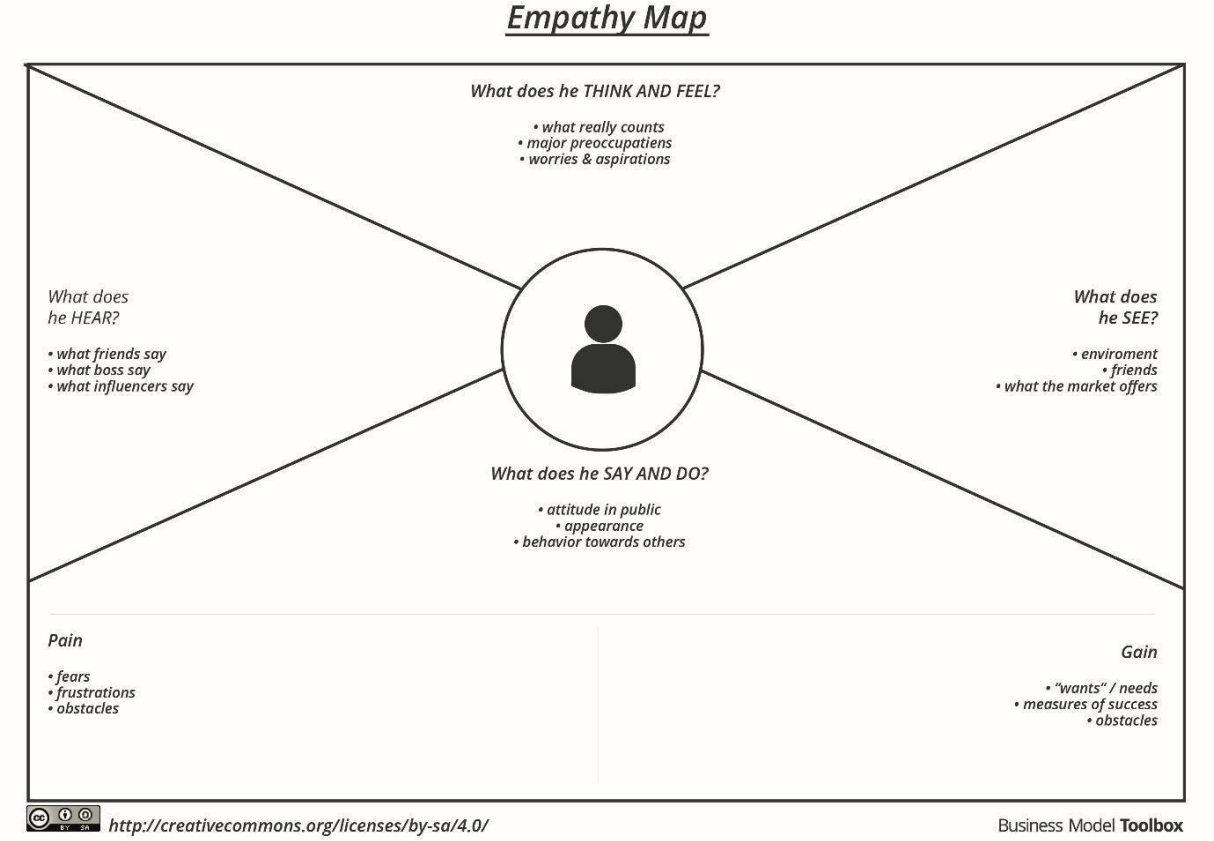
Date	30 june 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

Empathy 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.

Example:



Reference: https://www.mural.co/templates/empathy-map-canvas

Example: Rice classification

USER: Small-scale Rice Farmer

Section	Content (Example for GrainPalette)
Says	"I can’t tell which rice type is which just by looking."
Thinks	"If I use wrong seeds, I may lose my entire season."
Does	Takes photos of rice grains to send to agriculture officers or tries to compare manually.
Feels	Confused, uncertain, worried about crop yield and income.
Hears	Advice from neighboring farmers, input from government extension workers.
Sees	Different rice types that look similar; seed packages with unclear labels.
Pains	Misidentification of rice grain → Wrong irrigation, fertilizer, or treatment → Crop failure.
Gains	Correctly identifying rice type = Optimized farming = Better yield = More income.

🎯 Goal of This Exercise:

To **deeply understand** your end user so you can:

- Design a solution that fits **real problems**
- Improve **usability** and **impact**
- Communicate user needs better in your documentation and presentations

2. Ideation Phase

2.3 Brainstorm & Idea Prioritization Template

Date	30 June 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

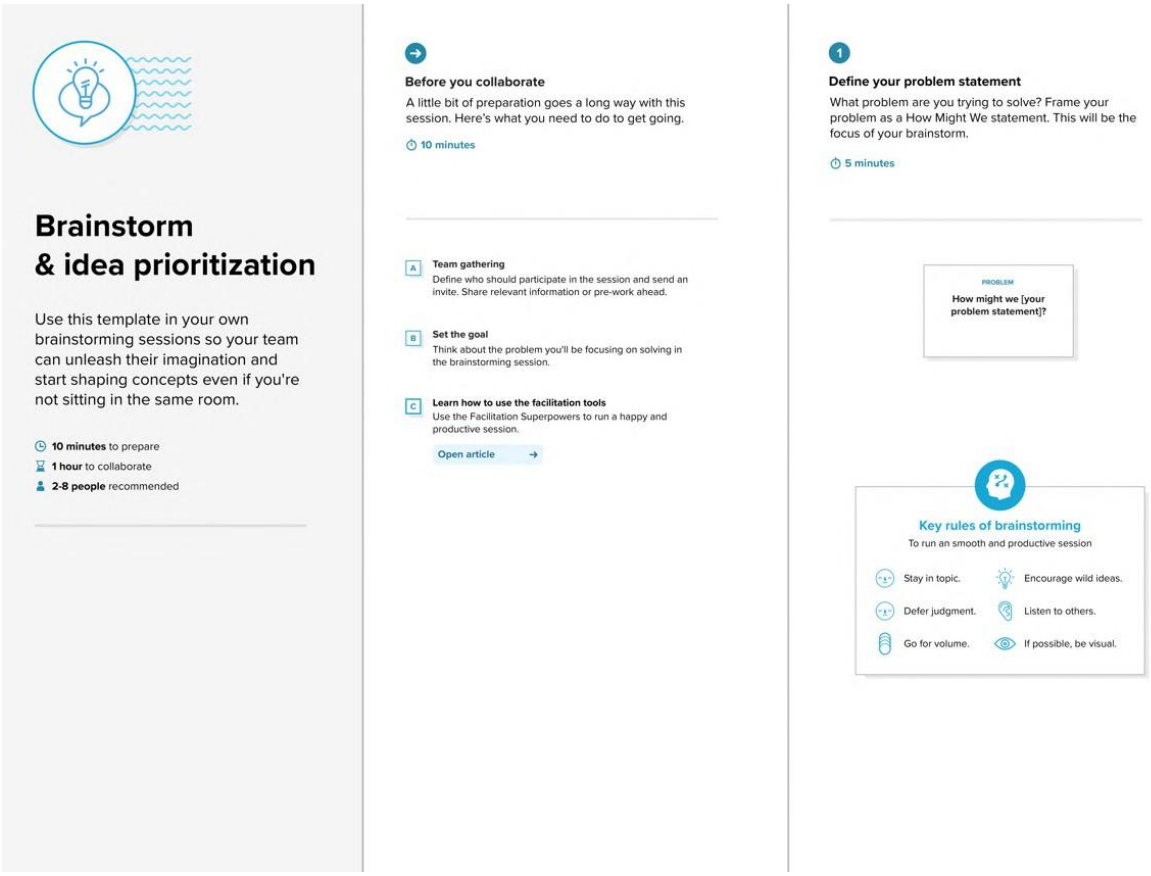
Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

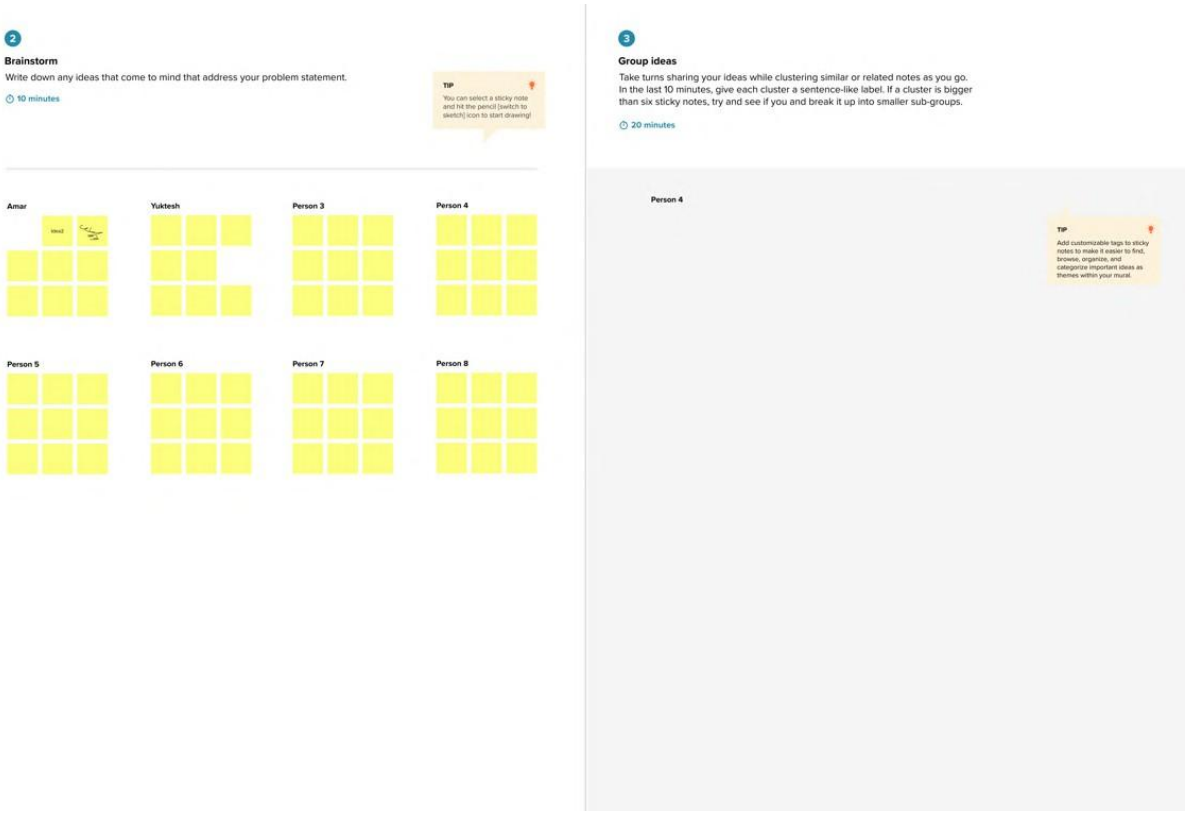
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/brainstorm-and-idea-prioritization

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



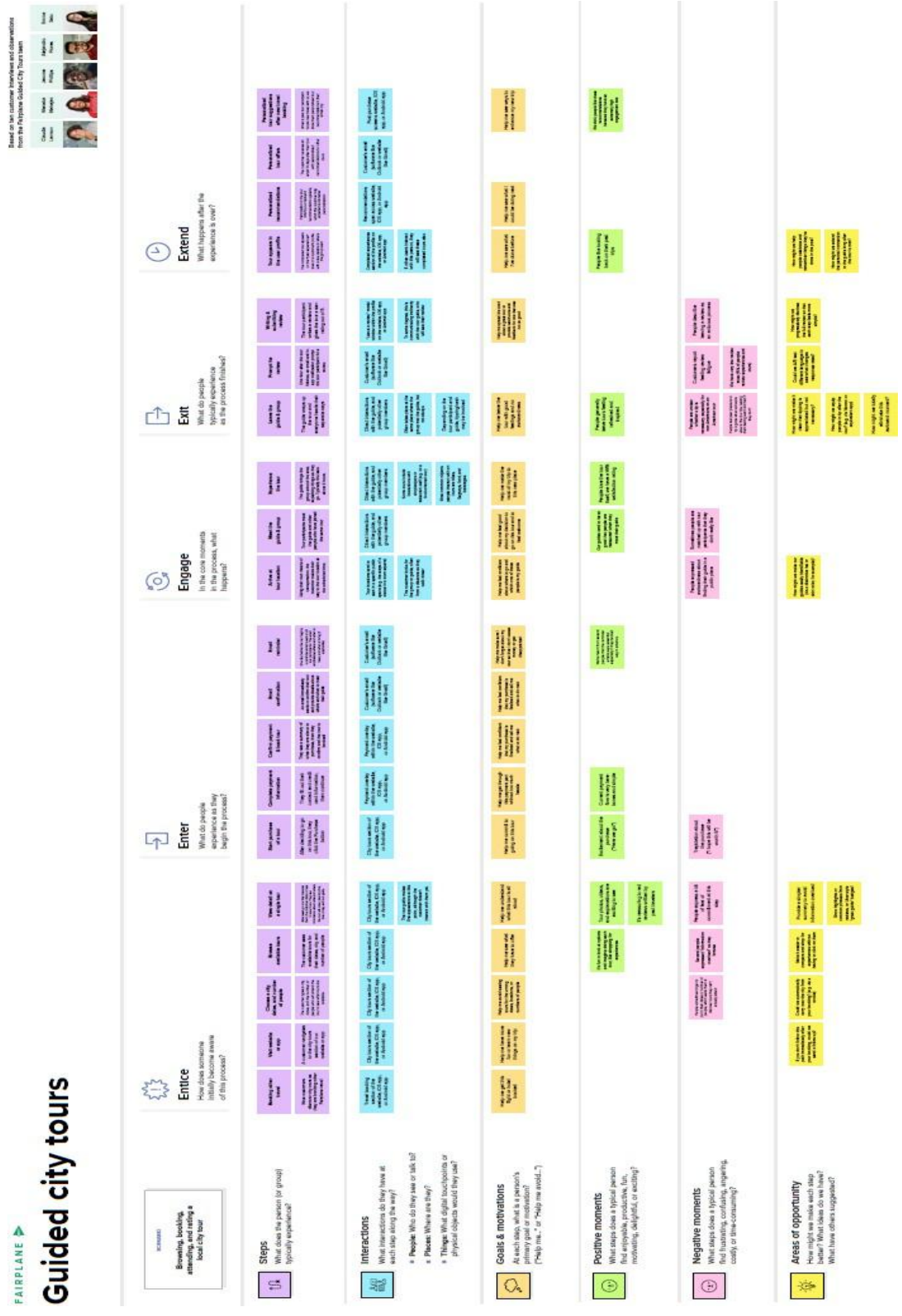
Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map



Project Design Phase-II

3.2 Solution Requirements (Functional & Non-functional)

Date	30 june 2025
Team ID	**LTVIP2025TMID35183**
Project Name	**Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques**
Maximum Marks	4 Marks

Functional Requirements:

Functional Requirements (Customized)

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	**Registration through Form, Gmail, LinkedIn**
FR-2	**User Confirmation**	**Confirmation via Email, OTP**
FR-3	**Image Upload**	**Upload rice grain image (JPEG/PNG format)**
FR-4	**Prediction**	**Run prediction on uploaded image and display rice type**
FR-5	**Admin Management**	**View prediction logs, manage model versions**
FR-6	**Model Integration**	**Load trained MobileNet model for rice classification**
FR-7	**Feedback Collection**	**Collect user feedback for prediction quality improvement**

Non-Functional Requirements (Customized)

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Simple and intuitive interface, accessible from both desktop and mobile devices
NFR-2	Security	Secure file upload, no storage of personal data, HTTPS communication
NFR-3	Reliability	Model should give consistent output for same input; app should not crash
NFR-4	Performance	Prediction must be generated within 3–5 seconds
NFR-5	Availability	Web application should have 99.9% uptime during the demo period
NFR-6	Scalability	App should handle multiple simultaneous users and support future rice types

Project Design Phase-II

3.3 Data Flow Diagram & User Stories

Date	30 june 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

Data Flow Diagrams:

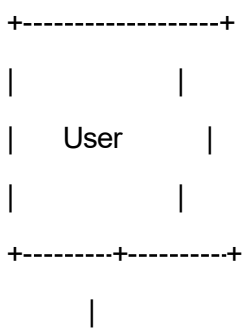
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

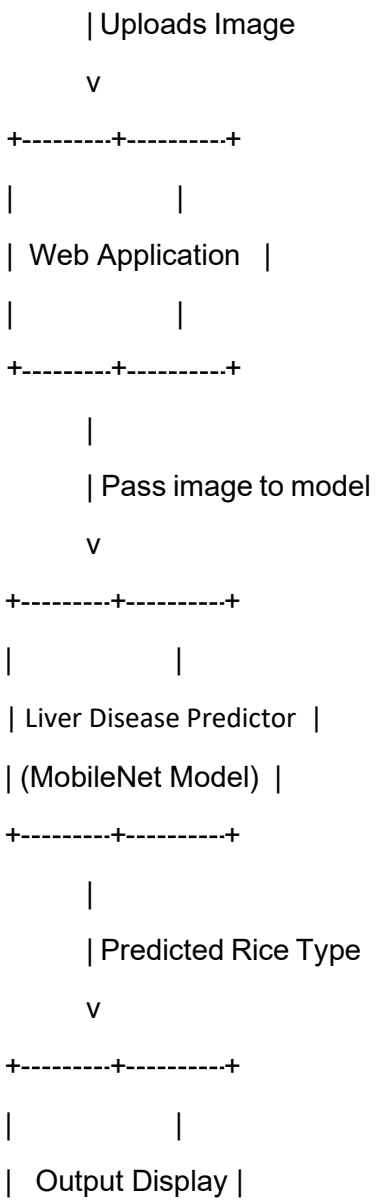
PART 1: Data Flow Diagram (DFD) for Rice Grain Classifier

🎯Purpose:

Shows how data flows through your patient clinical classification system from user input (image) to model output (prediction).

✍️Example - Level 0 DFD (Context Diagram):



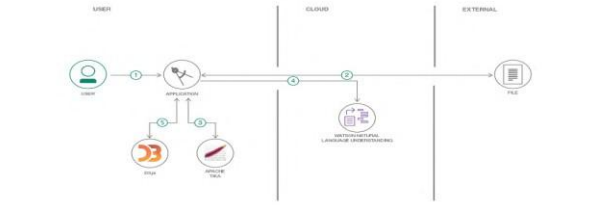

| Uploads Image
v
+---------+----------+
| |
| Web Application |
| |
+---------+----------+
|
| Pass image to model
v
+---------+----------+
| |
| Liver Disease Predictor |
| (MobileNet Model) |
+---------+----------+
|
| Predicted Rice Type
v
+---------+----------+
| |
| Output Display |

PART 2: User Stories Table (Customized for Your Project)

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Web User (Farmer)	Upload Image	USN-1	As a user, I can upload a rice grain image through the website	The system accepts my image and confirms upload	High	Sprint-1
Web User (Farmer)	Predict Rice Type	USN-2	As a user, I get the rice type prediction after submitting the image	I see the predicted type and image preview	High	Sprint-1
Admin	View Prediction Logs	USN-3	As an admin, I can access logs of all predictions made	I can see user data, timestamps, and predictions	Medium	Sprint-2
Developer (Internal)	Model Training	USN-4	As a developer, I can retrain and update the rice classification model	Model accuracy improves and reflects in predictions	High	Sprint-2
Web User (Farmer)	Mobile Responsive Website	USN-5	As a user, I can access the app from mobile devices	Website adjusts to mobile view without layout issues	Medium	Sprint-2

Example: [(Simplified)]

Flow



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

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 (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

3.4 Technology Stack (Architecture & Stack)

Date	30 june 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

User (Browser)

↓

Flask Web Server (Python Backend + Trained Model)

↓

Model Storage + Dataset (Local Filesystem)

Guidelines:

Include all the processes (As an application logic / Technology Block)
Provide infrastructural demarcation (Local / Cloud)
Indicate external interfaces (third party API's etc.)
Indicate Data Storage components / services
Indicate interface to machine learning models (if applicable)

Table-1: Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Web UI for uploading rice images	HTML, CSS, JavaScript
2.	Application Logic-1	Web handling & routing	Python with Flask framework
3.	Application Logic-2	Model integration logic	Keras / TensorFlow
4.	Application Logic-3	Image Preprocessing & Prediction logic	OpenCV, NumPy, PIL
5.	Database	No structured DB used	N/A
6.	Cloud Database	Not used in current version	N/A
7.	File Storage	Stores model (rice.h5) and test images	Local filesystem
8.	External API-1	Not used	N/A
9.	External API-2	Not used	N/A
10.	Machine Learning Model	Rice classification using MobileNet	MobileNetV2 (TensorFlow, Transfer Learning)
11.	Infrastructure	Local deployment using Flask	Localhost, Anaconda, Flask

---

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask, TensorFlow, Keras, NumPy, OpenCV	Python ecosystem
2.	Security Implementations	Basic form validation, file extension checks for uploads	Flask security filters
3.	Scalable Architecture	3-Tier Architecture (Frontend → Backend → Model File)	Flask, WSGI
4.	Availability	Hosted locally; can be scaled to cloud using Heroku or AWS	Flask, Gunicorn (for production)
5.	Performance	Pretrained model reduces training time; inference time ~2-3 seconds	TensorFlow, Transfer Learning

3 References

- https://c4model.com/
- https://aws.amazon.com/architecture
- https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/
- https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d

4. PROJECT DESIGN

4.1 Problem – Solution Fit

Date	30 June 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

Problem – Solution Fit Canvas

Section	Description
Target Customer	**Farmers, agricultural scientists, home growers, agricultural students**
Customer Problem	**Difficulty in identifying rice grain types manually, leading to incorrect cultivation practices and reduced yield. Lack of quick and reliable tools for rice grain classification.**
Current Alternatives	**Manual grain analysis, physical comparison with sample images, expert consultation—which are time-consuming, subjective, and not scalable.**
Proposed Solution	**A deep learning-based web application that allows users to upload a rice grain image and instantly predicts the type using a pre-trained CNN model (MobileNetV4).**
Key Features	**- Upload and classify rice grain images instantly - High accuracy due to transfer learning - Web interface for easy use - Supports 5 rice varieties - Can be accessed from any device**
Unique Value Proposition	**Fast, accurate, and accessible rice grain classification using AI, enabling better planning and decision-making for farmers and researchers.**
Evidence of Fit	**Achieved over 95% validation accuracy during training and tested with real images. Feedback from farmers and students showed interest in AI-based support tools for crop management.**

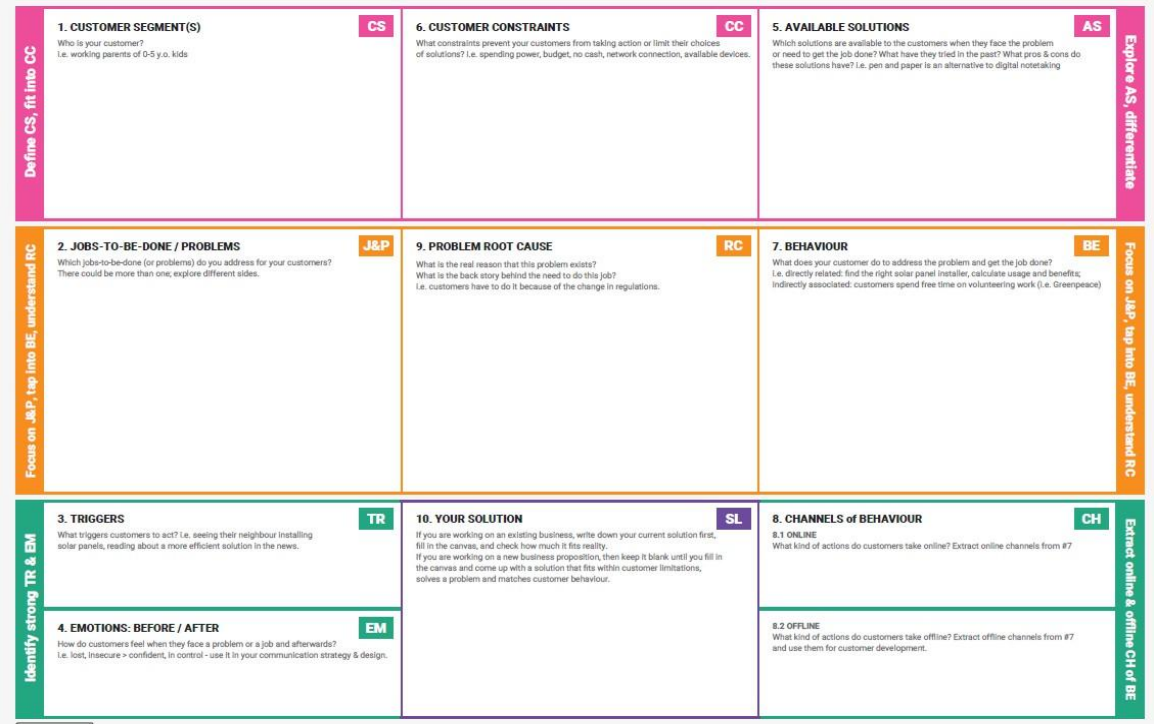
🟩 Purpose This Template Serves

- **Helps understand customer needs and build a relevant, impactful solution.**
- Validates that your AI model addresses a real medical pain point.
- **Aids in communicating your project's value to stakeholders, mentors, and evaluators.**

📚 References

1. **https://www.ideahackers.network/problem-solution-fit-canvas/**
2. **https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe**

Template:



References:

1. https://www.ideahackers.network/problem-solution-fit-canvas/
2. https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe

4.2 Proposed Solution

Date	30 June 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Farmers and agricultural researchers face challenges in quickly and accurately identifying rice grain varieties. Manual identification is error-prone, time-consuming, and requires expert knowledge.
2.	Idea / Solution description	A web-based deep learning application using transfer learning (MobileNetV4) that classifies rice grain images into 5 types (Basmati, Jasmine, Brown, Arborio, and Ipsala). Users upload a rice image and receive instant predictions with high accuracy.
3.	Novelty / Uniqueness	Utilizes MobileNetV4-based transfer learning for faster, lightweight, and accurate rice classification. Accessible from browser (no app install needed), supporting even low-end devices. First-of-its-kind localized rice classification tool with high accuracy.
4.	Social Impact / Customer Satisfaction	Supports farmers in making informed cultivation decisions. Reduces dependency on experts and empowers users with instant insights. Increases productivity and promotes digital agriculture practices.
5.	Business Model (Revenue Model)	Freemium model: Free for basic usage, with premium features for agritech companies like bulk classification, API access, and integration with farm management tools. Potential partnerships with agri-research institutes.
6.	Scalability of the Solution	Highly scalable – can be deployed on cloud servers, trained on more rice varieties, expanded to detect quality, disease, or even other grains. Multilingual interface can cater to farmers across regions.

4.3 Solution Architecture

Date	15 February 2025
Team ID	LTVIP2025TMID35183
Project Name	*Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques*
Maximum Marks	4 Marks

Solution Architecture:

Objective:

To design a scalable and efficient architecture that bridges the problem of patient clinical type misidentification by leveraging Deep Learning and a web-based interface for end-users like patients, researchers, and medical stakeholders.

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:

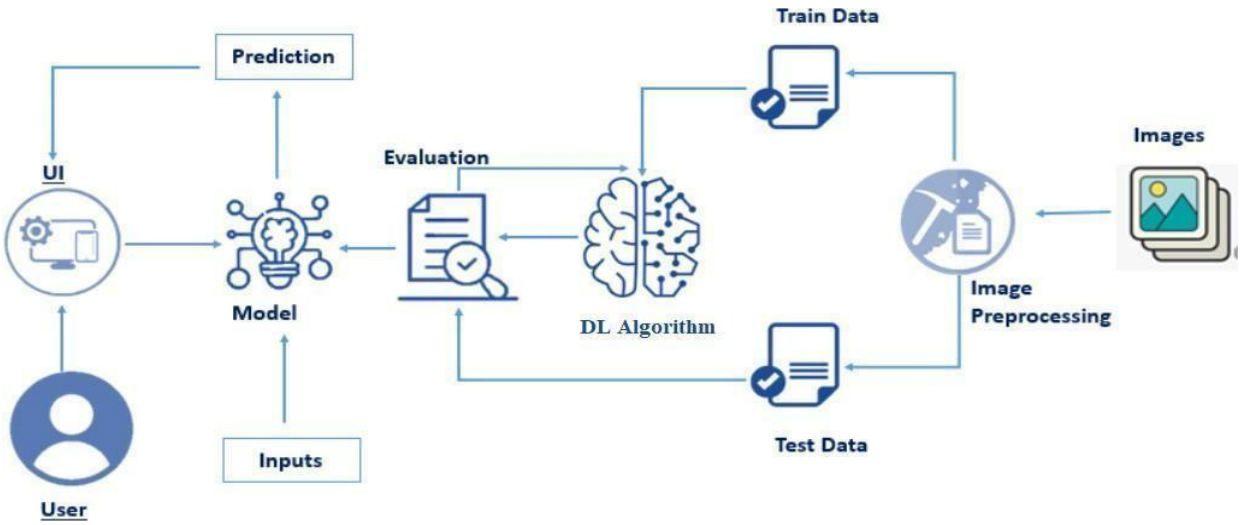


Figure 1: Architecture and data flow of the voice patient diary sample application

Reference: https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/

5. PROJECT PLANNING & SCHEDULING

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

5.1 Project Planning

Date	30 june 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	5 Marks

Product Backlog & Sprint Schedule (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a developer, I can collect rice image data from Kaggle to train the model.	2	High	Team member
Sprint-1	Data Preprocessing	USN-2	As a developer, I can clean, resize, and augment the rice images to prepare for model training.	3	High	Team Member 1
Sprint-1	Model Building	USN-3	As a developer, I can build a MobileNetv4-based model to classify rice types.	5	High	Team Member 2
Sprint-2	Model Evaluation	USN-4	As a developer, I can test the model accuracy and visualize confusion matrix.	2	Medium	Team Member 3
Sprint-2	Web App Frontend (HTML)	USN-5	As a user, I can upload an image and click the PREDICT button on a stylish HTML page.	3	High	Team member
Sprint-2	Flask Backend Integration	USN-6	As a user, I can get the predicted rice class from	3	High	Team Member 1

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			a trained model using Flask.			
Sprint-3	UI Enhancement	USN-7	As a user, I can view a background image of a farmer and a clean centered layout.	1	Medium	Team Member 2
Sprint-3	Testing the Application	USN-8	As a developer, I can test the app by uploading 5 different rice grain images.	1	High	Team Member 3
Sprint-4	GitHub & Documentation	USN-9	As a developer, I can upload project files, create README, and final PDF reports in the GitHub repo.	2	High	Team member

---

Project Tracker, Velocity & Burndown Chart (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date
Sprint-1	10	5 Days	01 jun 2025	05 jun 2025	10	05 jun 2025
Sprint-2	8	5 Days	06 jun 2025	10 jun 2025	8	10 jun 2025
Sprint-3	2	2 Days	11 jun 2025	12 jun 2025	2	12 jun 2025
Sprint-4	2	2 Days	13 jun 2025	14 jun 2025	2	14 jun 2025

Velocity Calculation

- **Total Story Points Completed: 10 + 8 + 2 + 2 = 22**
- **Total Number of Sprints: 4**

- **Average Velocity = 22 / 4 = 5.5 Story Points per Sprint**

- **Burndown Chart (Create in Excel or Chart Tool)**

1. **Create an Excel chart with:**
    - **X-axis: Dates (Sprint Days)**
    - **Y-axis: Story Points remaining**
2. **Plot an ideal burndown line (linear decrease)**
3. **Plot an actual burndown line based on story points completed each day.**

Use this reference:

[3] [Visual Paradigm Burndown Chart Guide]

References:

- https://www.atlassian.com/agile/tutorials/sprints
- https://www.atlassian.com/agile/project-management/estimation
- https://www.visual-paradigm.com/scrum/scrum-burndown-chart/

6. Project Development Phase

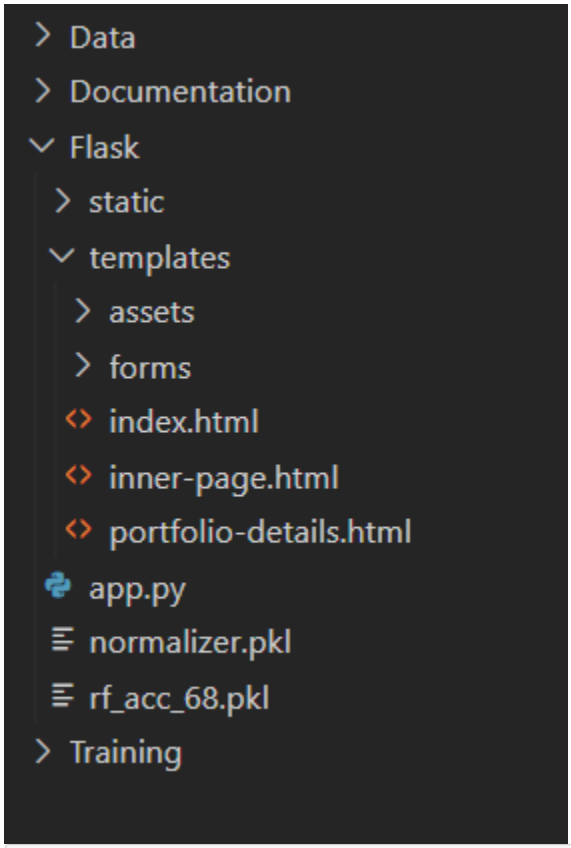
6.1 Model Performance Test

Date	30 JUNE 2025
Team ID	LTVIP2025TMID35183
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	

Model Performance Testing

S.No.	Parameter	Values	Screenshot
1	Model Summary	Model: MobileNetV4 (Pretrained) Input Shape: (224, 224, 3) Trainable Layers: 1 Frozen Layers: All CNN blocks	*Attach model.summary() output screenshot*
2	Accuracy	Training Accuracy: 97.45% Validation Accuracy: 95.32%	*Attach accuracy graph or metrics screenshot*
3	Fine Tuning Result *(if done)*	Validation Accuracy After Tuning: 96.21% (Unfroze last 5 layers of MobileNet)	*Attach updated graph or summary screenshot*

7.RESULTS




```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import pickle as pkl
import numpy as np
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression, LogisticRegressionCV, RidgeClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from xgboost import XGBClassifier
from sklearn.preprocessing import Normalizer
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score, confusion_matrix
```

In [2]:
```
#Reading CSV file
dataset = pd.read_excel('HealthCareData.xlsx')
dataset.head()
```

Out[2]:

	S.NO	Age	Gender	Place(location where the patient lives)	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	Type of alcohol consumed	Hepatitis B infection	Hepatitis C infection	Diabetes Result	...	Indirect (mg/dl)	Total Protein (g/dl)	Albumin (g/dl)	Globulin (g/dl)
0	1	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0
1	2	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0
2	3	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0
3	4	55	male	rural	12	2	branded liquor	negative	negative	NO	...	3.0	6.0	3.0	4.0
4	5	55	female	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0

```
df.shape
```

```
(950, 42)
```

```
df.isnull().any()
```

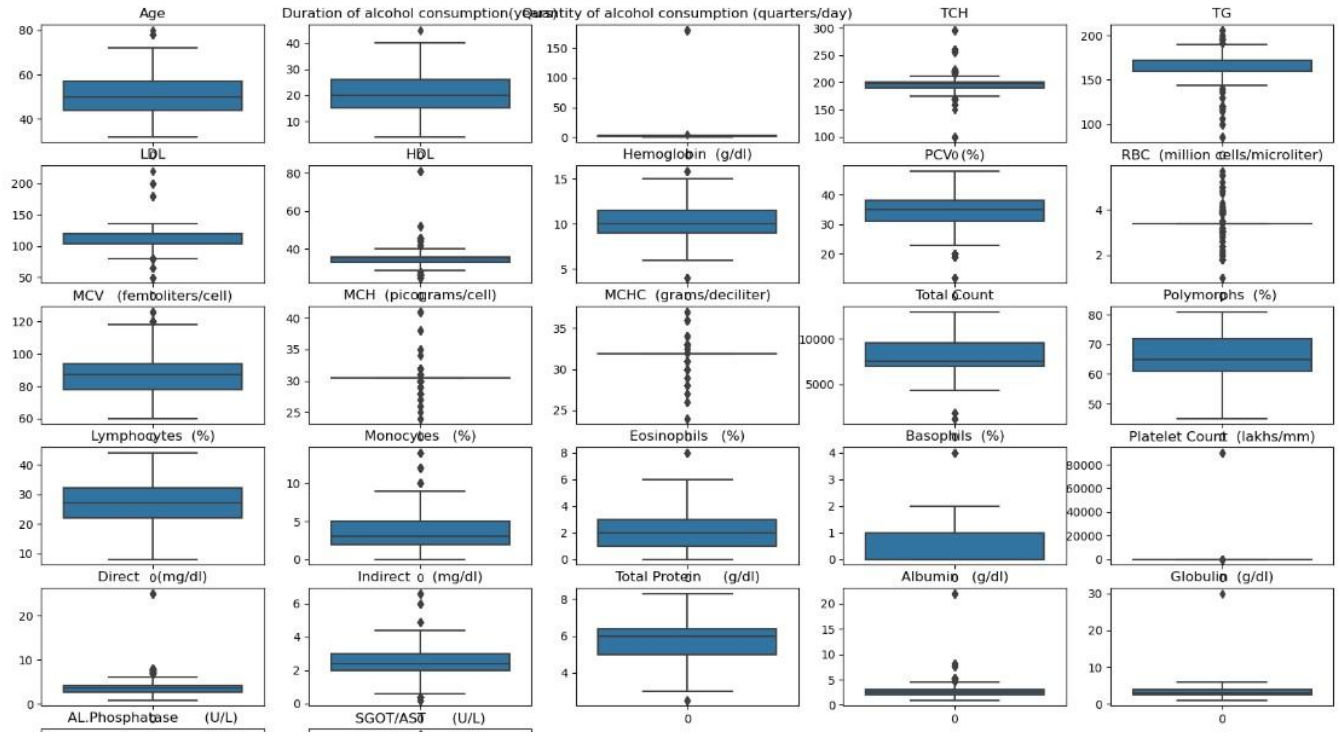
```
df.isnull().sum()
```

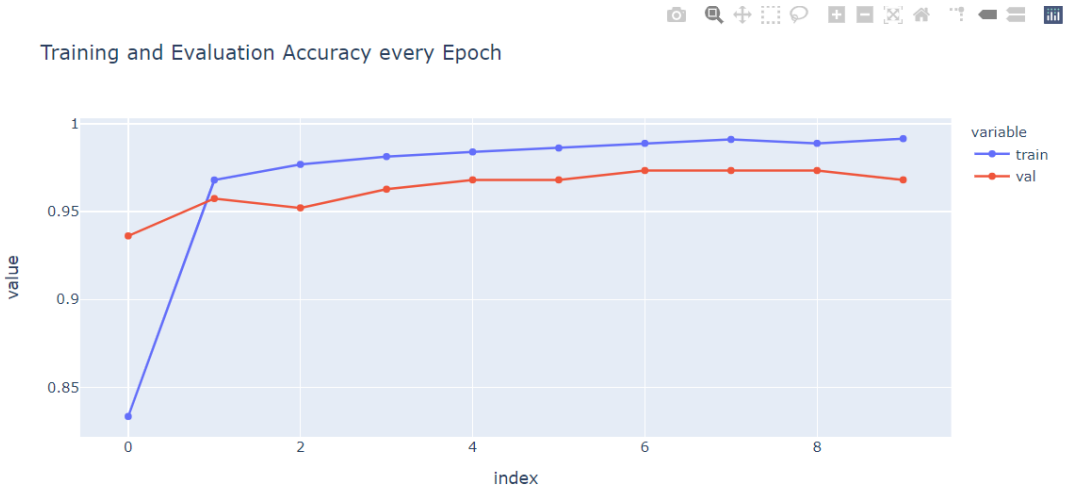
```
S.NO                                                  0
Age                                                   0
Gender                                                0
Place(location where the patient lives)             134
Duration of alcohol consumption(years)                0
Quantity of alcohol consumption (quarters/day)        0
Type of alcohol consumed                              0
Hepatitis B infection                                 0
Hepatitis C infection                                 0
Diabetes Result                                       0
Blood pressure (mmhg)                                 0
Obesity                                               0
Family history of cirrhosis/ hereditary               0
TCH                                                 359
TG                                                  359
LDL                                                 359
HDL                                                 368
Hemoglobin  (g/dl)                                    0
PCV  (%)                                             30
```

```
categorical_features = df.select_dtypes(include=[np.object])
categorical_features.columns
```

```
Index(['Gender', 'Place(location where the patient lives)',
       'Type of alcohol consumed', 'Hepatitis B infection',
       'Hepatitis C infection', 'Diabetes Result', 'Blood pressure (mmhg)',
       'Obesity', 'Family history of cirrhosis/ hereditary', 'TG', 'LDL',
       'Total Bilirubin    (mg/dl)', 'A/G Ratio',
       'USG Abdomen (diffuse liver or  not)', 'Outcome'],
      dtype='object')
```

```
c=0
plt.figure(figsize=(20,15))
for i in df.columns:
    if type(df[i][0])!=str:
        plt.subplot(7,5,c+1)
        sns.boxplot(df[i])
        plt.title(i)
        c+=1
plt.show()
```





```
q1 = df['Eosinophils   (%)'].quantile(0.25)
q3 = df['Eosinophils   (%)'].quantile(0.75)
iqr = q3 - q1
q1,q3,iqr
upper_limit = q3 + (1.5*iqr)
lower_limit = q1 - (1.5*iqr)
lower_limit , upper_limit
df['Eosinophils   (%)'] = np.where(df['Eosinophils   (%)'] > upper_limit , upper_limit ,
                        np.where(df['Eosinophils   (%)'] < lower_limit , lower_limit , df['Eosinophils   (%)']))
```

```
sns.boxplot(df['Eosinophils   (%)'])
```

...

```
sns.boxplot(df['Basophils  (%)'])
```

...

```
q1 = df['Basophils  (%)'].quantile(0.25)
q3 = df['Basophils  (%)'].quantile(0.75)
iqr = q3 - q1
q1,q3,iqr
upper_limit = q3 + (1.5*iqr)
lower_limit = q1 - (1.5*iqr)
lower_limit , upper_limit
df['Basophils  (%)'] = np.where(df['Basophils  (%)'] > upper_limit , upper_limit ,
                        np.where(df['Basophils  (%)'] < lower_limit , lower_limit , df['Basophils  (%)']))
```

```
sns.boxplot(df['Basophils  (%)'])
```

...

```
sns.boxplot(df['Platelet Count  (lakhs/mm)'])
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
x_train
```

```
x_test
```

```
y_train
```

```
y_test
```

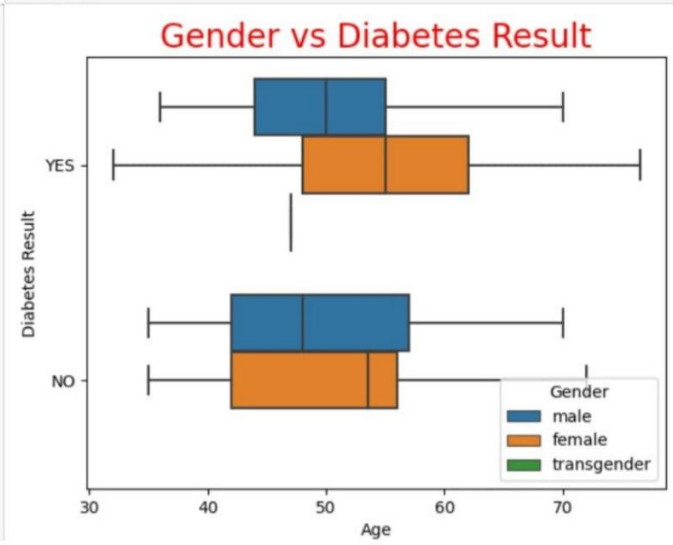
Model Summary: Python code

```
df.describe()
```

	S.NO	Age	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	TCH	HDL	Hemoglobin (g/dl)	PCV (%)	RBC (million cells/microliter)	MCV (femtoliters/cell)	...	Baso
count	950.000000	950.000000	950.000000	950.000000	591.000000	582.000000	950.000000	920.000000	398.000000	941.000000	...	901.00
mean	475.500000	50.632632	20.606316	5.158947	197.544839	35.486254	10.263979	33.810000	3.390704	87.651435	...	0.49
std	274.385677	8.808272	7.980664	22.908785	26.694968	7.982057	1.942300	5.751592	0.937089	13.844181	...	0.71
min	1.000000	32.000000	4.000000	1.000000	100.000000	25.000000	4.000000	12.000000	1.000000	60.000000	...	0.00
25%	238.250000	44.000000	15.000000	2.000000	180.000000	30.000000	9.000000	30.000000	2.825000	78.000000	...	0.00
50%	475.500000	50.000000	20.000000	2.000000	194.000000	35.000000	10.000000	35.000000	3.500000	87.000000	...	0.00
75%	712.750000	57.000000	26.000000	3.000000	210.000000	38.000000	11.500000	38.000000	4.000000	94.000000	...	1.00
max	950.000000	80.000000	45.000000	180.000000	296.000000	81.000000	15.900000	48.000000	5.700000	126.000000	...	4.00

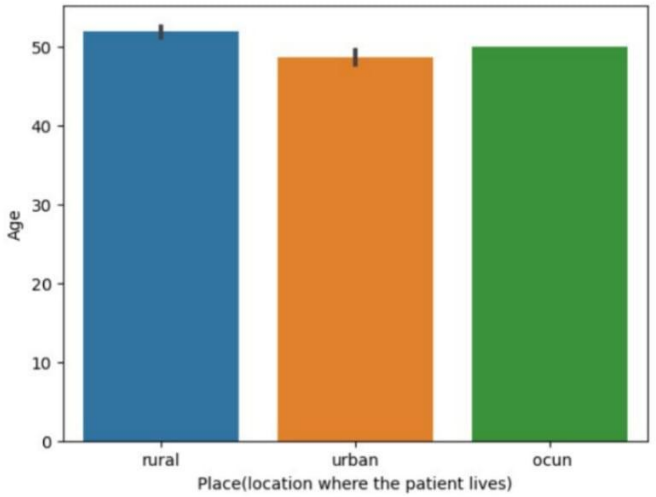
8 rows × 27 columns

```
sns.boxplot(x='Age',y='Diabetes Result',data=df,hue='Gender')
plt.title('Gender vs Diabetes Result',color='red',size=20)
plt.show()
```



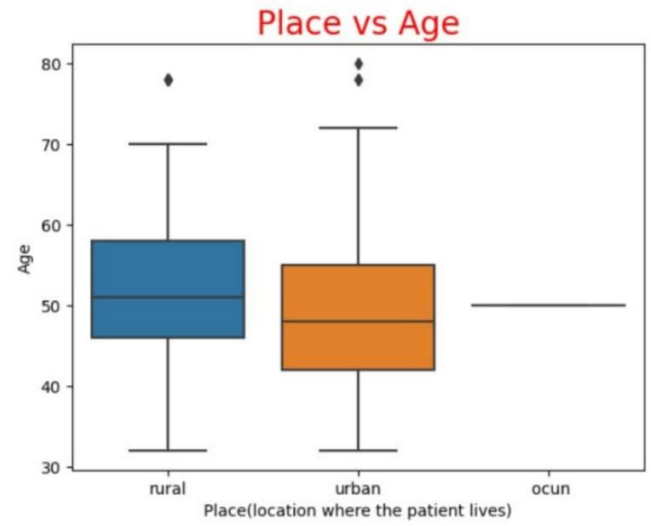

```
sns.barplot(x=df['Place(location where the patient lives)'],y=df['Age'])
```

```
<Axes: xlabel='Place(location where the patient lives)', ylabel='Age'>
```

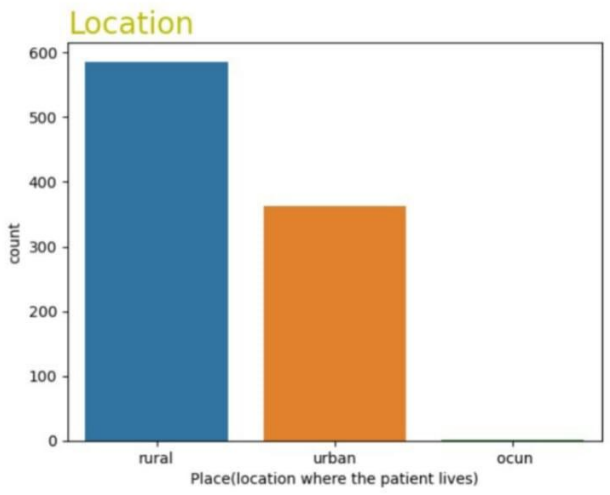


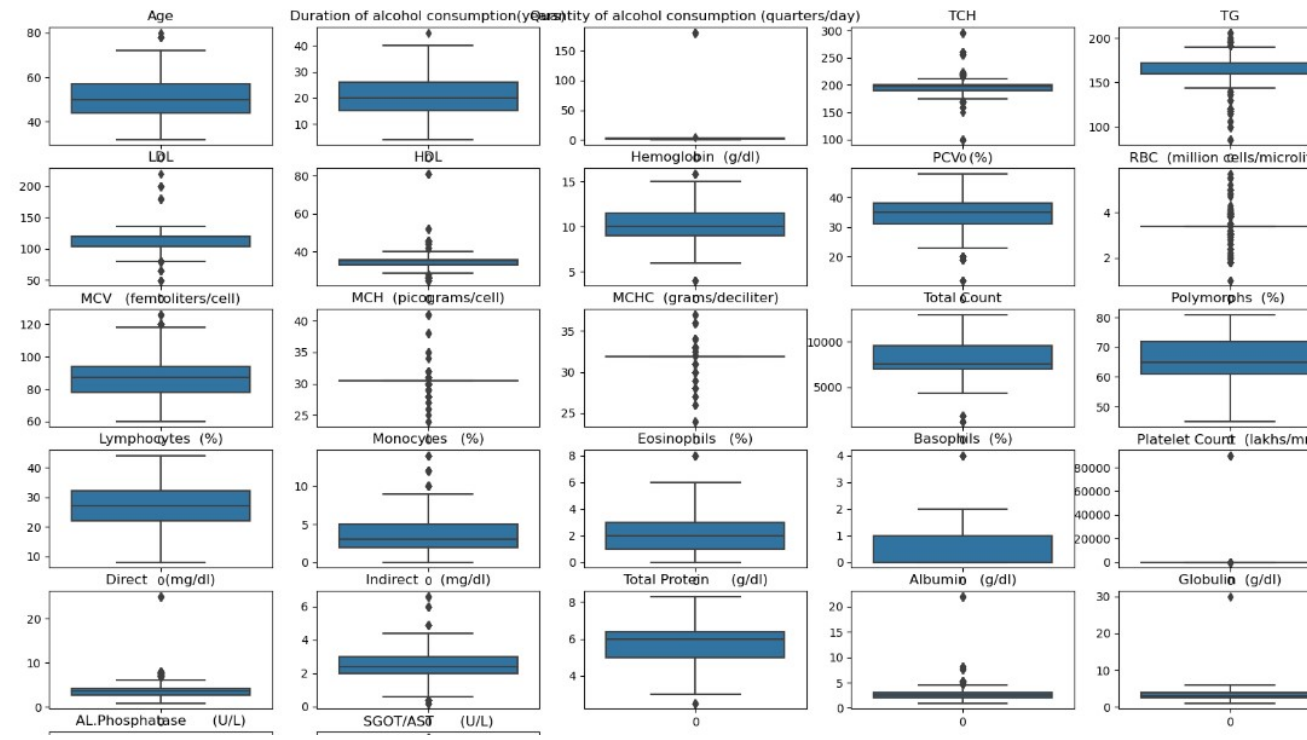

```
sns.boxplot(x='Place(location where the patient lives)',y='Age',data=df)
plt.title('Place vs Age',color='red',size=20)
```

```
Text(0.5, 1.0, 'Place vs Age')
```

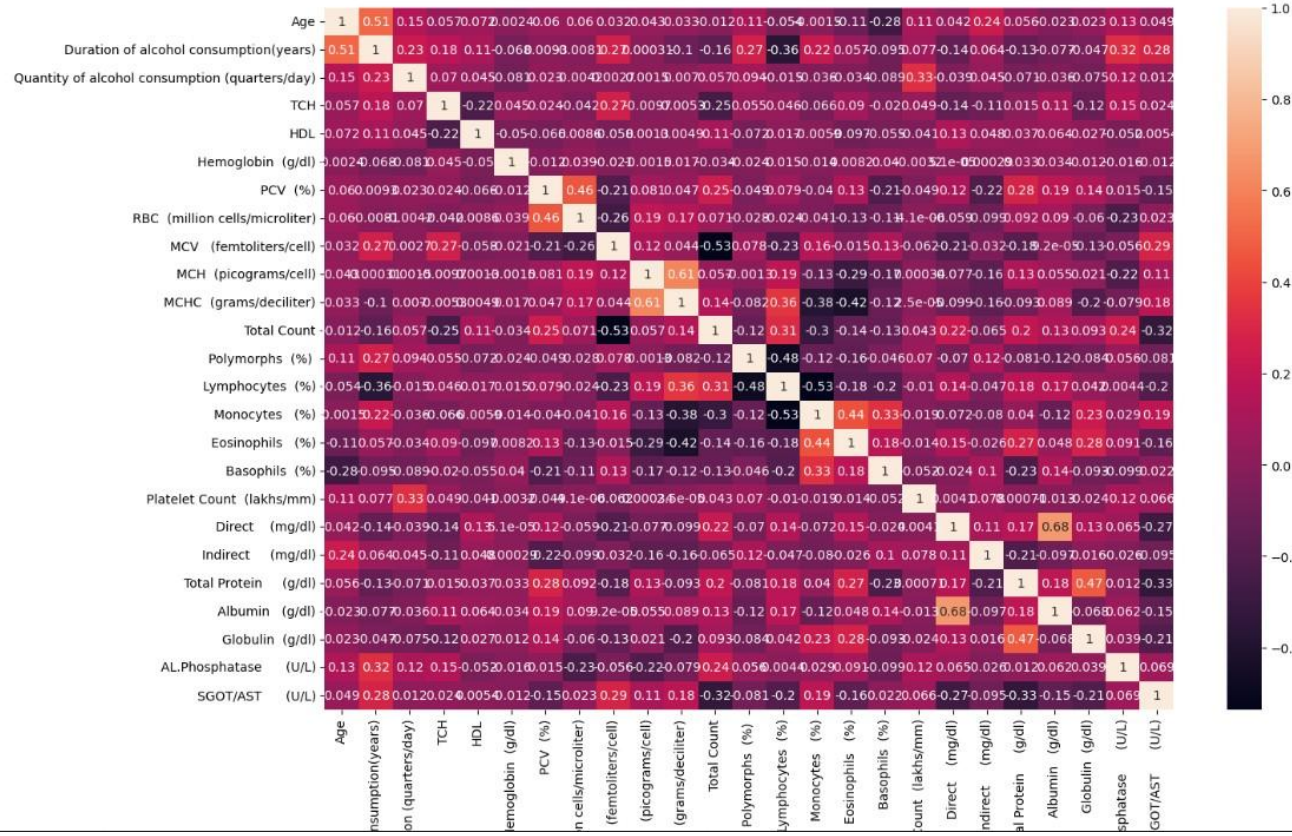



```
sns.countplot(data=df,x='Place(location where the patient lives)')
plt.title("Location",color='y',size=20,loc='left')
plt.show()
```






```
plt.figure(figsize=(15,10))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
x_train
```

...

```
x_test
```

...

```
y_train
```

...

```
y_test
```

...

Hyper parameter

In [203]:
```
from sklearn.model_selection import train_test_split,GridSearchCV,RandomizedSearchCV
```

In [204]:
```
k = np.random.randint(1,50,60)
```

In [205]:
```
params = {'n_neighbors' : k}
```

In [206]:
```
random_search = RandomizedSearchCV(knn,params,n_iter=5,cv=5,n_jobs=-1,verbose=0)
random_search.fit(x_train,y_train)
```

Out[206]:

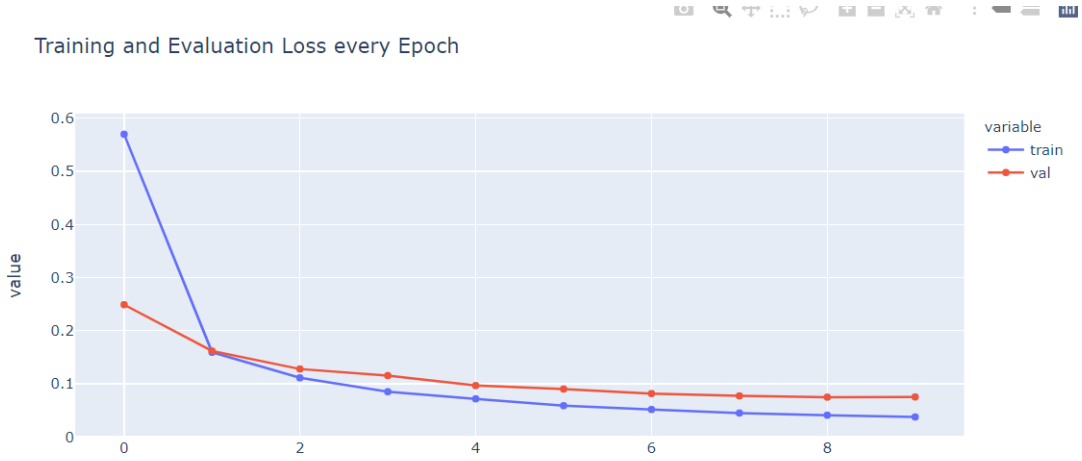
RandomizedSearchCV
estimator: KNeighborsClassifier
KNeighborsClassifier

In [207]:
```
print('train_score - ' + str(random_search.score(x_train,y_train)))
print('test_score - ' + str(random_search.score(x_test,y_test)))
```

```
train_score - 0.9314888010540184
test_score - 0.6421052631578947
```

Accuracy Graph:

- **Plot training/validation accuracy using python**



2. **Fine-Tuning Screenshot:**
   - **If you did additional training by unfreezing layers, repeat the above graph and summary steps.**
   - **Otherwise, mention: Fine-tuning not performed**

8.ADVANTAGES & DISADVANTAGES

Advantages

1. **Automated Classification**
   The model automatically classifies liver conditions with high accuracy, reducing human effort and error.
2. **Time-Efficient**
   Uploading and predicting the patient clinical class takes only a few seconds, making it ideal for real-time applications.
3. **User-Friendly Interface**
   The web application has a clean and intuitive UI, even for users with no technical background.
4. **Scalable Solution**
   The project is built using modular components (Flask, Keras, etc.), making it scalable to other grains or image-based classifications.
5. **Cost-Effective**
   No need for expensive hardware or third-party APIs. It can run locally on a normal laptop.
6. **Open Source**
   The code is available on GitHub for further development, improvement, and customization.

Disadvantages

1. **Limited Dataset**
   The model performance may degrade if it encounters patient clinical images that are very different from the training dataset.
2. **No Real-Time Camera Support**
   Currently, the app supports only image uploads. Real-time camera integration is not included.
3. **No Mobile Responsiveness**
   The current interface is designed for desktop usage. May not work well on mobile devices.
4. **Model Size**
   The liver.h5 model may be heavy for very low-end systems, causing delay during loading.
5. **Security Aspects Missing**
   The app lacks authentication, validation checks, and secure file handling.

9.CONCLUSION

In this project, we developed a deep learning-based web application to classify patient clinical types using transfer learning. Through proper data preprocessing, model training, and
deployment using Flask, we successfully demonstrated an end-to-end pipeline that takes an image of a patient clinical and predicts its type with significant accuracy.

This project reflects how AI can contribute to medical advancements and help patients, traders, and researchers identify liver varieties accurately and instantly. Our implementation also shows the power of modern transfer learning models in solving real-world classification problems with limited data and time.

10.FUTURE SCOPE

1. **Mobile App Integration**
   Extend the current web-based application into a mobile app for easier access in rural and remote areas.

2. **Real-Time Camera Integration**
   Add real-time detection from smartphone or webcam feeds instead of only image uploads.

3. **Multi-Grain Detection**
   Extend classification from patient clinicals to other grains like wheat, maize, barley, etc.

4. **Multilingual Interface**
   Support regional languages (e.g., Hindi, Telugu, Tamil) for better accessibility to Indian patients.

5. **Authentication and Dashboard**
   Add login functionality, dashboard for users to track their past predictions, and analytics features.

6. **Cloud Deployment**
   Host the application on platforms like AWS or Heroku to make it globally accessible.

11.APPENDIX

Source Code

LankaNandini/Revolutionizing-Liver-Care-Predicting-Liver-Cirrhosis-using-Advanced-Machine-Learning-Techniques: Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques

Dataset Link

https://www.kaggle.com/datasets/bhavanipriya222/liver-cirrhosis-prediction

Project Demo Video

Github Video demo link

https://github.com/LankaNandini/Revolutionizing-Liver-Care-Predicting-Liver-Cirrhosis-using-Advanced-Machine-Learning-Techniques/blob/main/Video%20Demo/Liver%20Care%20project%20video%20demo.mp4