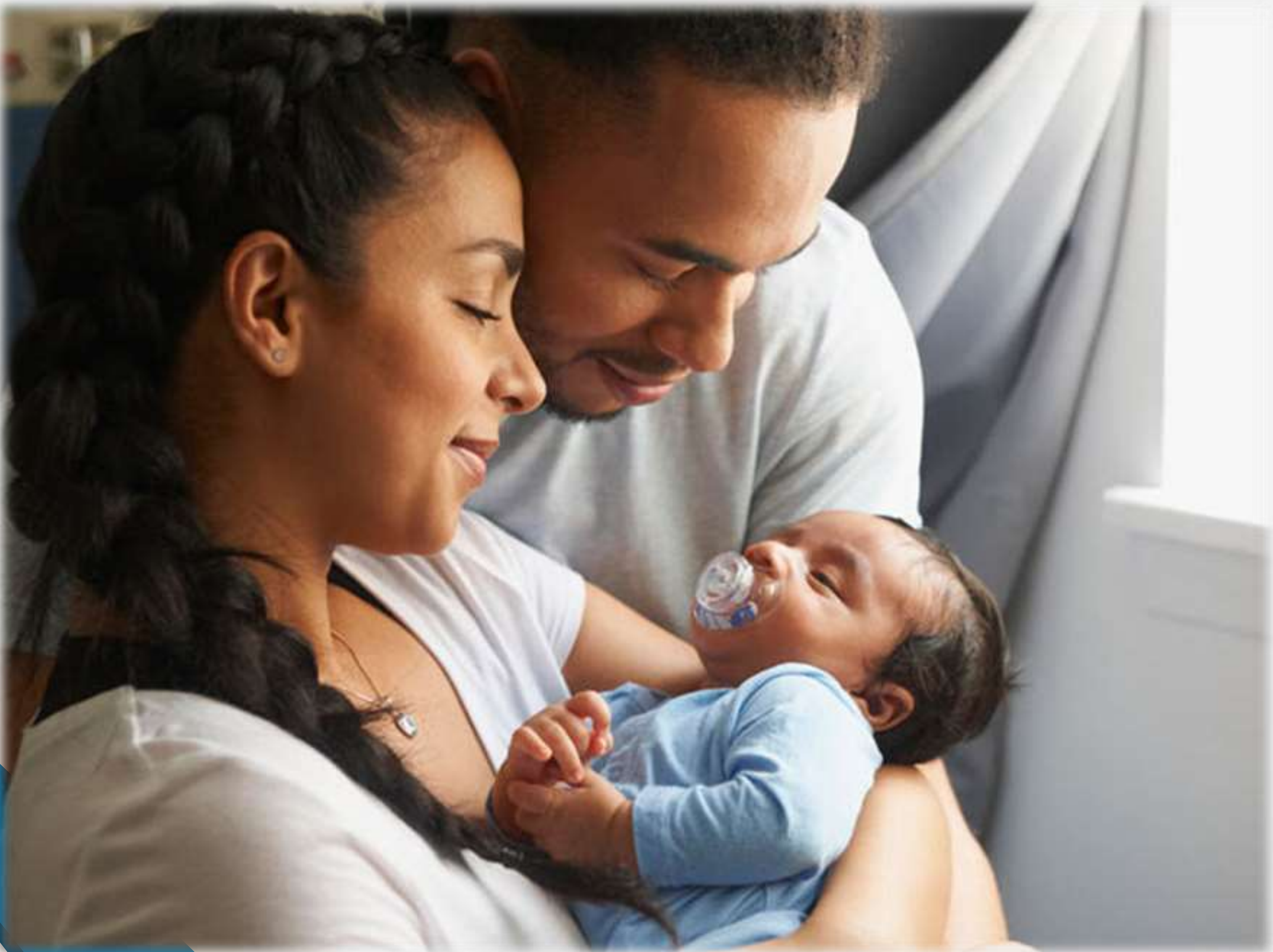
A MECHANISM TO KEEP THE IMMUNIZATION HIGH OF NEWBORNS WITH CRUCIAL TECHNIQUES VIA REMOTE CONSULTATION

(TMP-23-103)



Progress after proposal presentation

- Changes of the components
- Information gathering
  - ✓ Field visits
  - ✓ Interviews
  - ✓ Research papers, tutorials
- Implementations (basic models)

Changes of the components

Component Name	Before Proposal presentation	After Proposal presentation
Decentralized patient information system	• Permission-based access	• Permission-based access control to ensure that only authorized parties can access the data. • Improving Novellty
Chatbot Application	• Medical knowledge base prediction for diseases • Suggestions of remedies • System notifications for the vaccinations • Chat history	• Identify skin infections using image processing • As the novel feature, uploading an image option was introduced by the panel

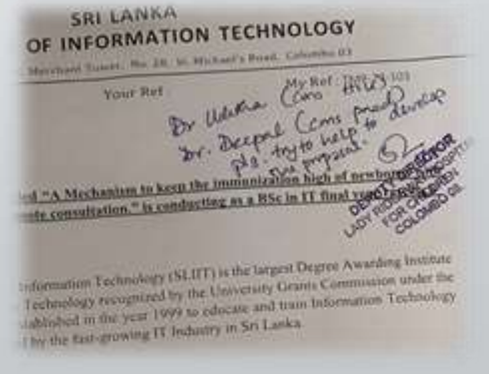
Changes of the components

Component Name	Before Proposal presentation	After Proposal presentation
Nutrition Level and malnutrition detection	• Prediction of nutritional level using image processing and calculations. • Predict the nutritional level using height, weight, and head circumference	• Predict the growth level without using images but behavioral aspects • Component name changed to "**Growth Level Prediction**"
Infant Sickness Identification through Video Processing	• Remote consultation between the pediatrician and the parents with infants. • Predict the sickness and suggest the solutions and remedies.	• Video processing participate only parents with the infants • Only identify the sickness and abnormal behavior and notify it through the system

Information Gathering

Interviews

- **Deputy Director - Lady Ridgeway Hospital**
- **Dr. Uditha - IT Consultant (LRH)**
- **Dr. Deepal - Pediatrition (LRH)**
- **The pediatric clinic was visited at LRH**
- **Visited the Castle Hospital and handed over the necessary documentation for approval to do field visits**



SRI LANKA
OF INFORMATION TECHNOLOGY
Merchant Tower, No. 28, St. Michael's Road, Colombo 03

Your Ref

My Ref: [illegible] 101

Dr Uditha (Cons [illegible])
Dr. Deepal (Cons paed)
pls. try to help to develop
the proposal.

...ed "A Mechanism to keep the immunization high of newborn[illegible]
...ode consultation." is conducting as a BSc in IT final year [illegible]

DEPUTY DIRECTOR
LADY RIDGEWAY HOSPITAL
FOR CHILDREN
COLOMBO 08.

...Information Technology (SLIIT) is the largest Degree Awarding Institute ... Technology recognized by the University Grants Commission under the ... established in the year 1999 to educate and train Information Technology ... by the fast-growing IT Industry in Sri Lanka.

IT20015244 | Wijesinghe R.M.U.S

Decentralized Patient Registration.

Software Engineering

Background and Literature

- I implement the system to allow parents and health care providers to manage a child's medical and personal information in an efficient and accessible manner. For example, if a child needs to see a specialist, the specialist can quickly access the child's medical history and provide the necessary treatment.
- And if you want to view the patient records, you need to request the public key from the patient.
- I would implement a permission-based access control mechanism to ensure that only authorized parties can access the data.
- By encapsulating all of these features into a single component, the DPR component could be easily integrated into a larger decentralized healthcare system.

Background and Literature cont.

- In this application is powered by IPFS, where patients' medical records are stored on the distributed file system, not owned by any centralized entity. A patient can access his or her records by interacting with a smart contract on the Ethereum blockchain, forming a digital identity of the patient on the decentralized network.
- The client first connects with MetaMask, and uses smart contract to mint a patient or doctor block, registered by the wallet address.
- The client can upload a record file to IPFS, which address is linked to a patient block in ETH chain. The client can get all record addressed stored in a patient block from smart contract, and get a record file by its address from IPFS.
- The health provider can search for a patient's records using the address, and upload a new record for the patient. The patient can also view his or her records.
- A patient or a doctor can access the patient's records by interacting with a smart contract on the Ethereum blockchain.

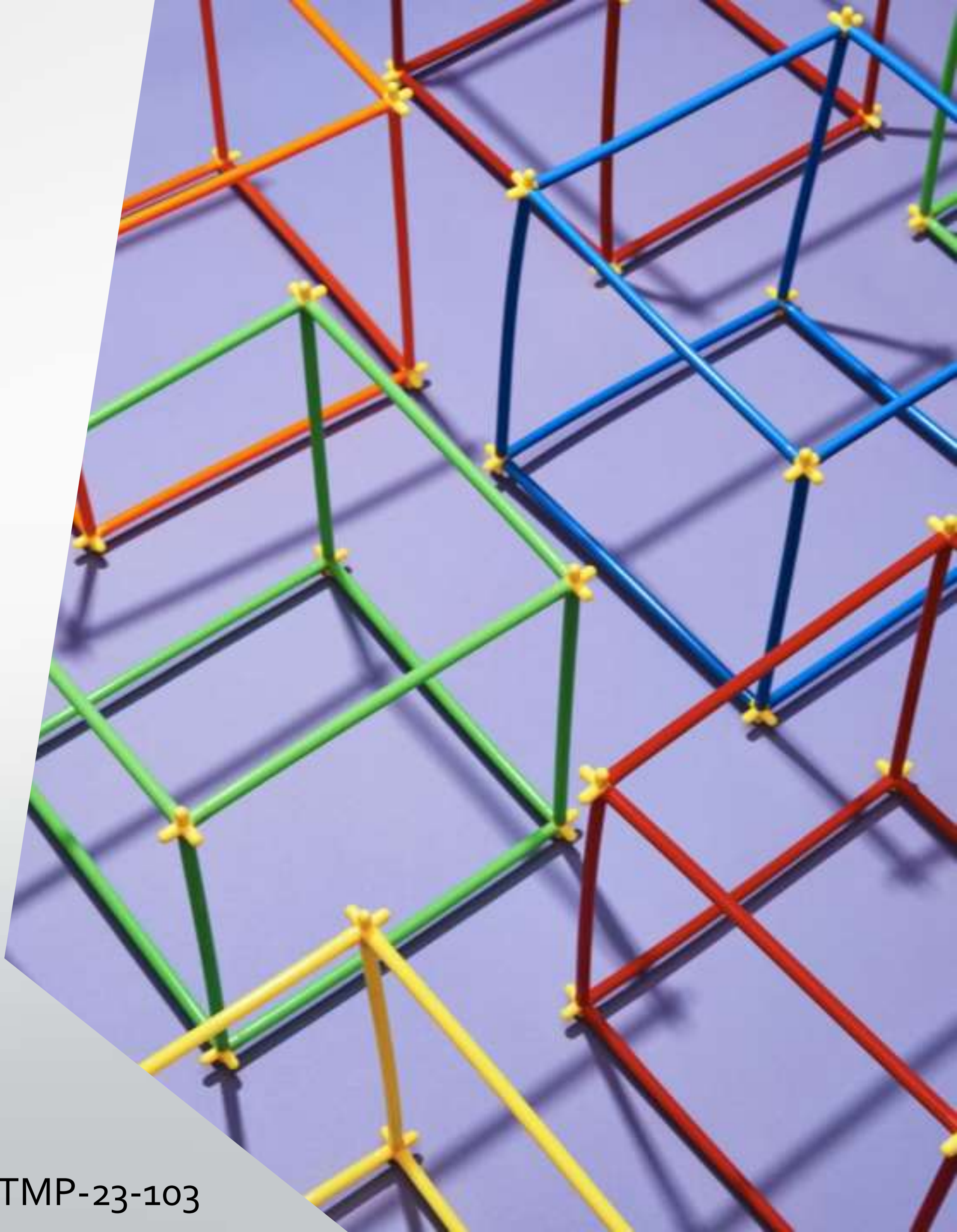
Specific Objectives & sub-Objectives

Specific Objective

The goal of using a decentralized system for the registration of newborns and parents' details is to provide a secure, transparent, and efficient way to manage critical information related to newborns and their parents.

Sub Objectives

- By leveraging decentralized technologies such as blockchain or decentralized databases, this system can enhance security and privacy, prevent data tampering, and improve data accessibility.
- The smart contract stores the newborns' and parents' details on the Ethereum blockchain, ensuring that the data is secure and immutable.
- This system also allows parents and healthcare providers to manage the child's medical and personal information in a transparent and secure way, while providing access to relevant parties as needed.

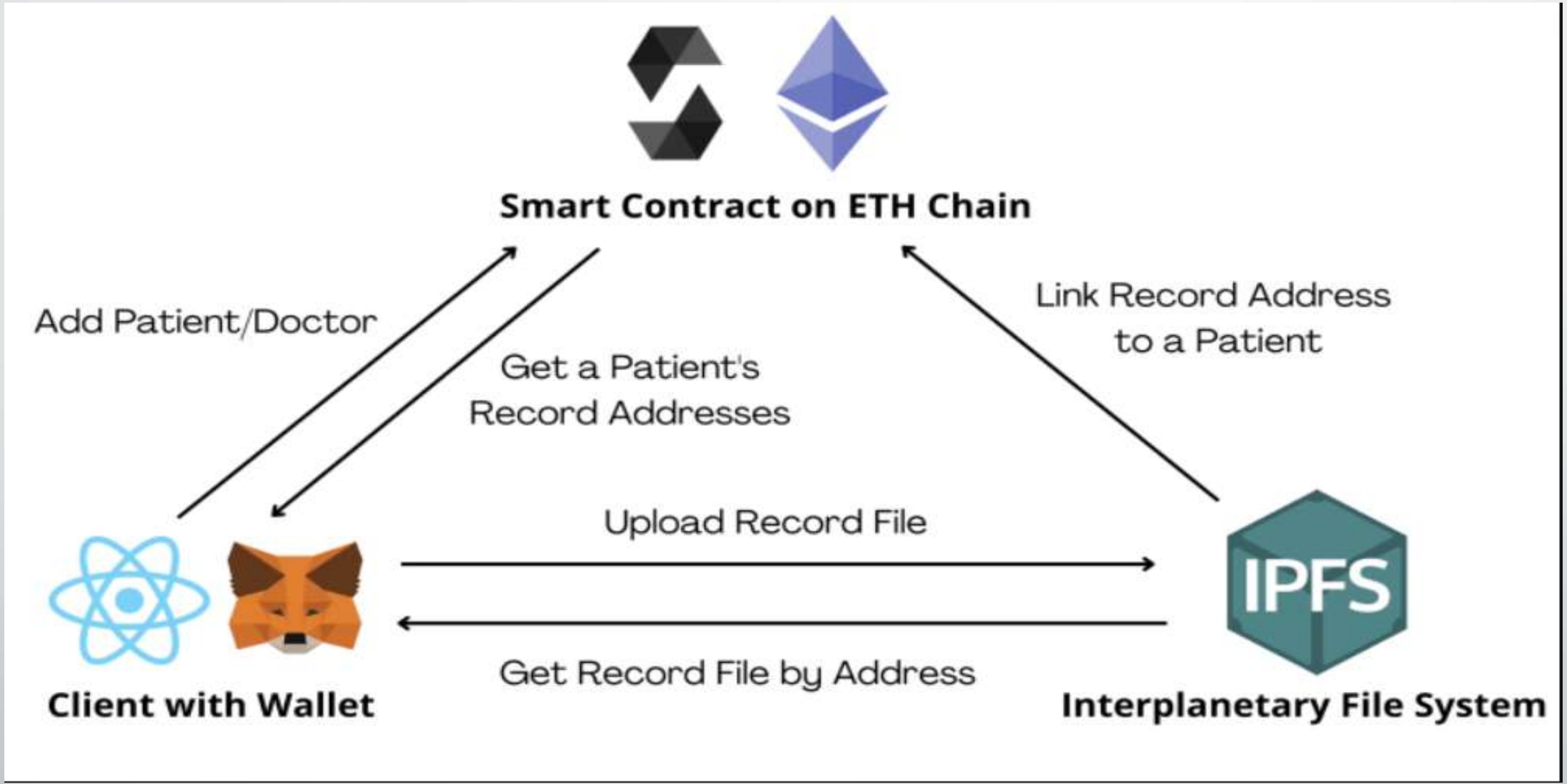


Methodology

Component overview Diagram

Technology Involved

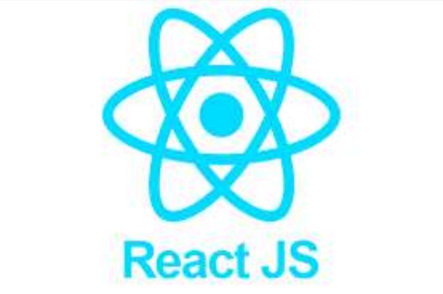
Component Overview Diagram


Smart Contract on ETH Chain
Add Patient/Doctor
Get a Patient's Record Addresses
Link Record Address to a Patient
Upload Record File
Get Record File by Address
IPFS
Client with Wallet
Interplanetary File System

Technology Involved

Frontend

- **Reactjs, HTML, CSS, Tailwind.**



Backend

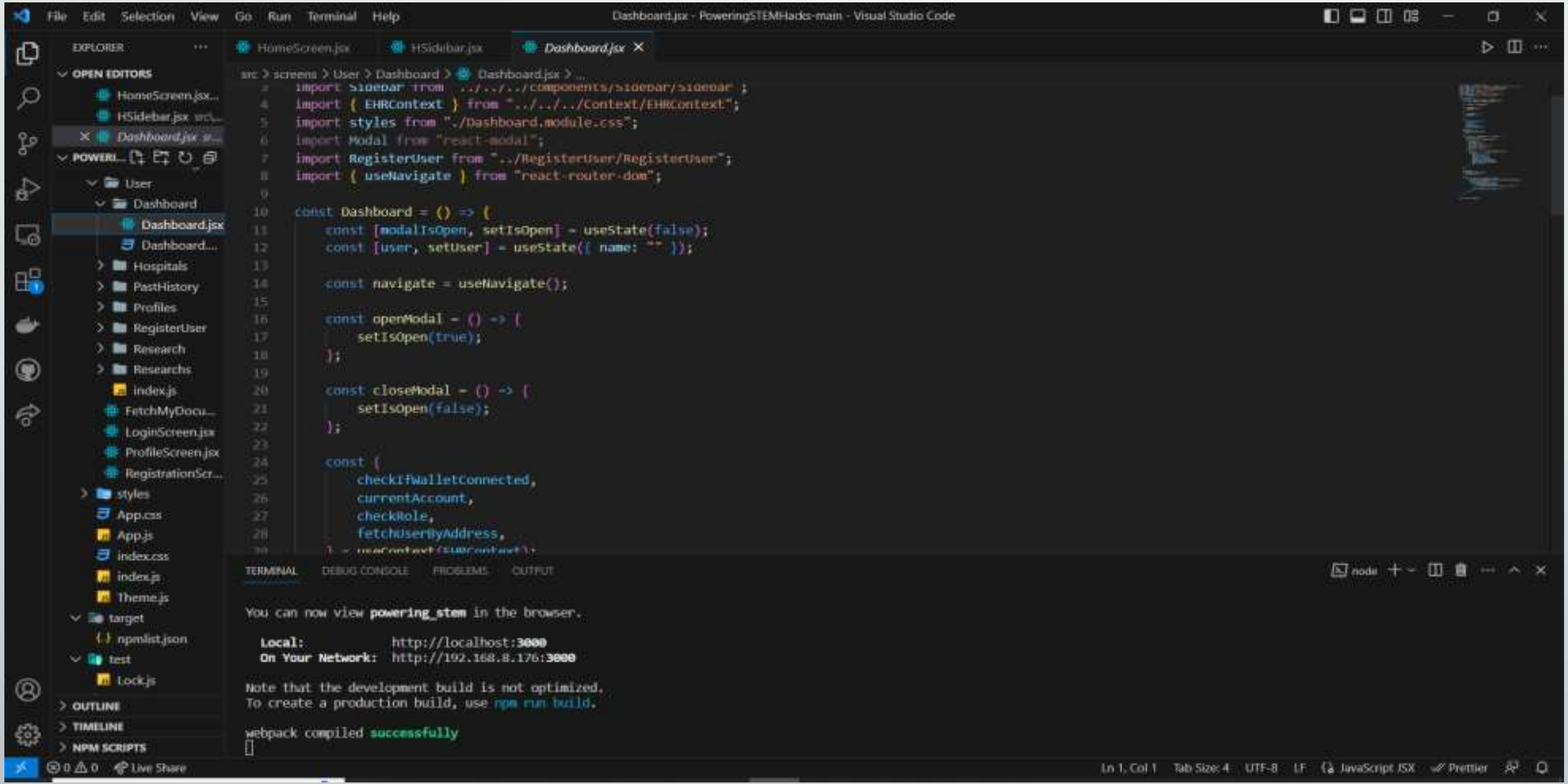
- Solidity and Ethereum Based Blockchain
- Hardhat

Database

- **Web3.Storage,, Inter Planetary File System(IPFS)**

Developed solution

User Dashboard Code

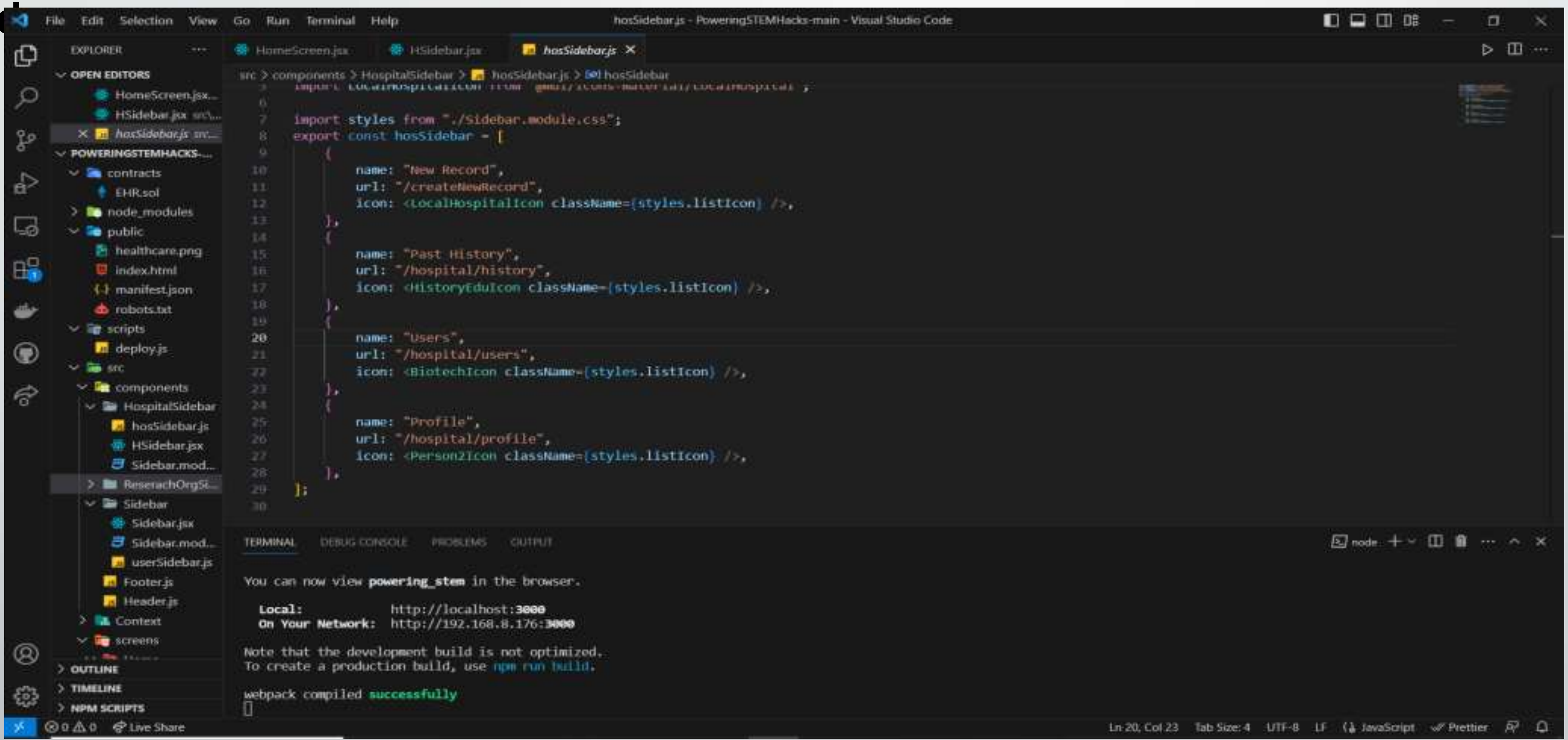


```
import Sidebar from "../../../components/Sidebar/Sidebar";
import { EHRContext } from "../../../Context/EHRContext";
import styles from "./Dashboard.module.css";
import Modal from "react-modal";
import RegisterUser from "../RegisterUser/RegisterUser";
import { useNavigate } from "react-router-dom";

const Dashboard = () => {
    const [modalIsOpen, setIsOpen] = useState(false);
    const [user, setUser] = useState({ name: "" });

    const navigate = useNavigate();

    const openModal = () => {
        setIsOpen(true);
    };

    const closeModal = () => {
        setIsOpen(false);
    };

    const {
        checkIfWalletConnected,
        currentAccount,
        checkRole,
        fetchUserByAddress,
    } = useContext(EHRContext);
```

```
You can now view powering_stem in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.8.176:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Developed solution cont.

Hospital Dashboard Co

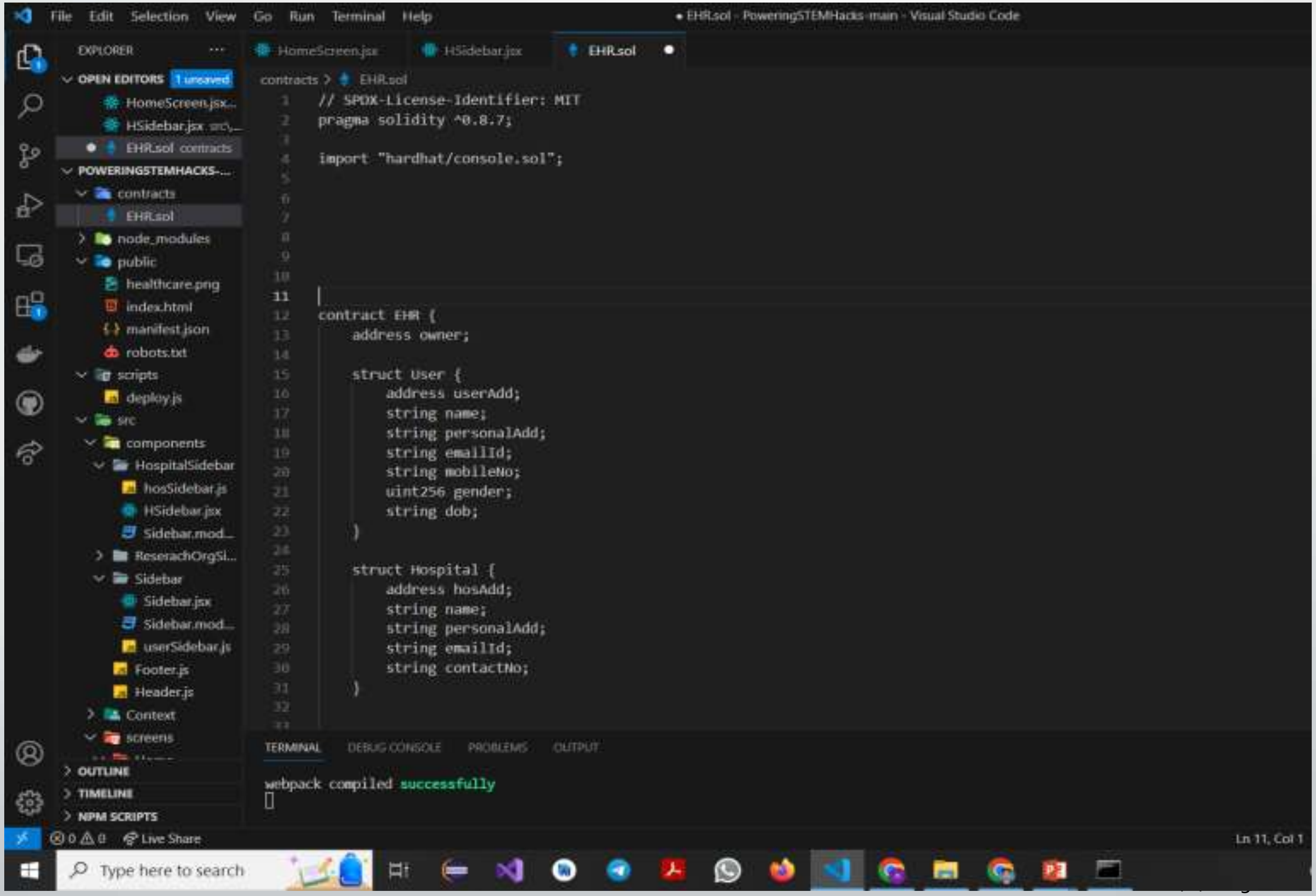


```javascript
import LocalHospitalIcon from "@mui/icons-material/LocalHospital";

import styles from "./Sidebar.module.css";
export const hosSidebar = [
    {
        name: "New Record",
        url: "/createNewRecord",
        icon: <LocalHospitalIcon className={styles.listIcon} />,
    },
    {
        name: "Past History",
        url: "/hospital/history",
        icon: <HistoryEduIcon className={styles.listIcon} />,
    },
    {
        name: "Users",
        url: "/hospital/users",
        icon: <BiotechIcon className={styles.listIcon} />,
    },
    {
        name: "Profile",
        url: "/hospital/profile",
        icon: <Person2Icon className={styles.listIcon} />,
    },
];
```

```
You can now view powering_stem in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.8.176:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Developed solution

Smart Contract code

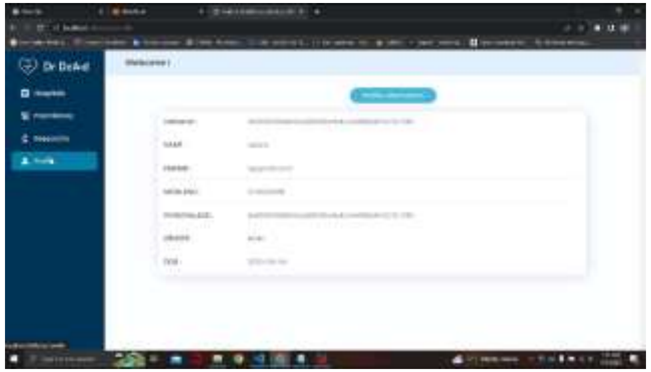


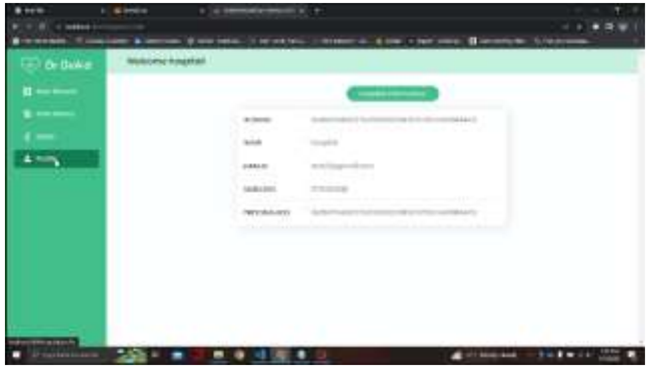
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

import "hardhat/console.sol";








contract EHR {
    address owner;

    struct User {
        address userAdd;
        string name;
        string personalAdd;
        string emailId;
        string mobileNo;
        uint256 gender;
        string dob;
    }

    struct Hospital {
        address hosAdd;
        string name;
        string personalAdd;
        string emailId;
        string contactNo;
    }

```

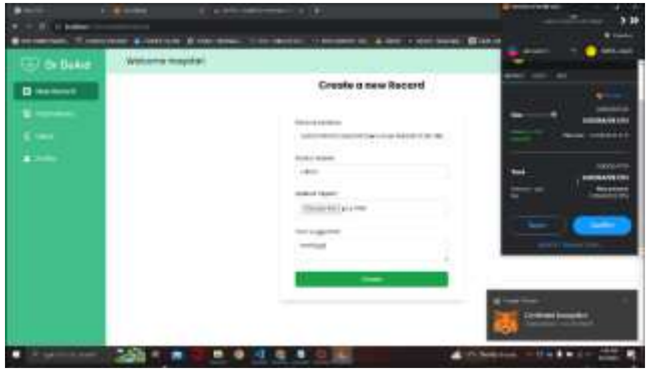
```
webpack compiled successfully
```

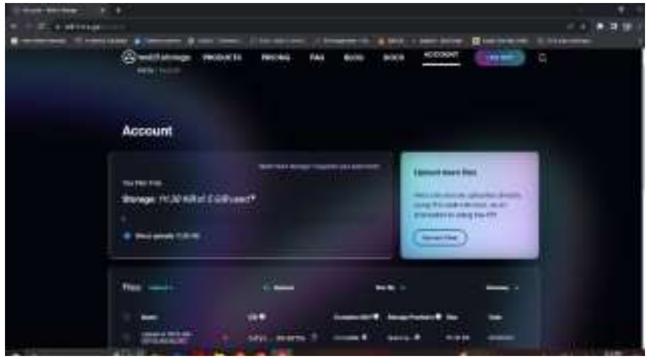
15

Results and Discussion









Project Requirements

- ## Functional requirements

- Distributed storage: The system must be able to store data in a distributed manner, with multiple copies of data spread across nodes in the network.

- Decentralized consensus: The system must use a consensus algorithm to agree on the state of the network without relying on a central authority. This could be achieved using a blockchain, DAG, or other consensus mechanisms.

- P2P communication: The system must be able to facilitate communication between nodes without relying on a centralized server. This could be achieved using a peer-to-peer (P2P) network.

- User authentication and access control: The system must provide secure authentication and access control mechanisms to ensure that only authorized users can access and modify data

- ## Non-Functional requirements

Accuracy, Reliability, Performance, Usability , Accessibility,

References

1. Secure decentralized electronic health records sharing system based on blockchains: https://www.frontiersin.org/articles/10.3389/fbloc.2021.732112/full

2. Secure decentralized electronic health records sharing system based on blockchains: https://www.sciencedirect.com/science/article/pii/S1319157821001051.

3. C C Darshan Thimmaiah, D. S. (2019). Decentralized Electronic Medical Records. IJRAR, 7.

4. Catherine Quantin, G. C. (2009). Centralised versus Decentralised Management of Patients' Medical Records. researchgate, 6.

5. Jihui Shi 1, S. K. (2022). A Novel Block Chain Method for Urban Digitization. International Journal of Environmental, 19.

6. Syed Agha Hassnain Mohsan, A. R. (2021). Decentralized Patient-Centric Report and Medical Image. International Journal of Environmental, 18.

7. Zhang P, White J, Schmidt DC, Lenz G and Rosenbloom (2018), FHIR Chain: Applying Blockchain to Securely and Scalably Share, Computational and Structural Biotechnology Journal, 16: 267-278

8. Crypt Bytes Tech (2017), Medical Chain - A blockchain for electronic health records. https:// medium.com/crypt-bytes-tech/medicalchain-a-blockchain-for-electronic-health-recordseef181ed14c2.

9. Ekblaw A, Azaria A, Halamka JD and Lippman A (2016), A Case Study for Blockchain in Healthcare: “MedRec” prototype for electronic health records and medical research data. MIT Media Lab

IT20147778| Jayawickrama N.D.D

Health Informatics Chatbot Application

Software Engineering

Background and Literature

- **Introduction and Background**
- The proposed component is to assist the parents of babies who are having a hard time managing their time.
- The system enables parents to ask questions about their infants' behaviors, any medicine, related remedies, or any other assumption or methods of how exactly to react to an emergency situation.
- By the image processing technique, the chatbot can identify the skin infections of the kid by analyzing a clear uploaded photo.
- Chat bot will also notify the vaccination days according to the data records it gets from the decentralized system.
- From this specific component, the main target is to introduce an assistant that is available beyond the time, helps to manage the time, to reduce the unnecessary cost of doctor appointments, and assist the parents in their problematic situations as a trained health care agent.
- **User Workflow**
- Parent approach to the chatbot provided the problematic situation they have faced.
- The chatbot system takes the input as data, asks relevant questions to understand the situation, and comes to a final prediction.
- Then the suggestions will be displayed as an answer to the problem. This suggestion can be advice, a prediction of a disease, a history record, a remedy, a medicine, or a contact detail of a relevant doctor.

Background and Literature cont.

- **Benefits for Parents**
- Time and cost Management: This system will reduce the unnecessary amount of time and the cost for the travel to hospital.
- Early Detection: By capturing and analyzing the parents' data, the system can predict the sicknesses and suggest solutions.
- Availability: Unlike doctors, the system is available for the user without any appointments and without time consideration.
- De-stress mental health: System existence is to decrease the problematic and emerging situation and to smoothly handle it.

- **Impact and Future Scope**
- The system touches the sensitive area of the health industry by predicting the sickness of infants.
- This method can be recognized as remote consultation via telecommunication, and with the reliability of the prediction, the system may perform an important role as an assistant for the parents of the newborn child.
- Through the training model that contributes with AI and ML algorithms, Image processing techniques, and NLP processing techniques, the accuracy, reliability, and user-friendliness o the system can be optimized.

Information Gathering

- **Number of interview sessions were conducted to finalize the component by the field specialists.**

✓ Dr. Christine Buddhini – Panadura Hospital

✓ Dr. Thilini Amarathunga - Muttur Hospital

✓ Dr. Deepal – LRH hospital

- **Research was conducted after the comments of the panel and the specialists.**
  - **Existing systems**
  - **Research papers**
  - **Video tutorials**
- **For the implementation, tutorial videos, blogs, and chatbot documentation are followed.**

Research Problem

Chatbot Application for Assisting and Guiding Infant Parents

User Interaction and Engagement: How can you design the chatbot to effectively engage with parents and provide them with relevant information and advice regarding infant care?

Content and Information: What kind of information should the chatbot provide to parents? How can you ensure the information is accurate and up-to-date?

Skin Rash Disease Identification from Uploaded Images

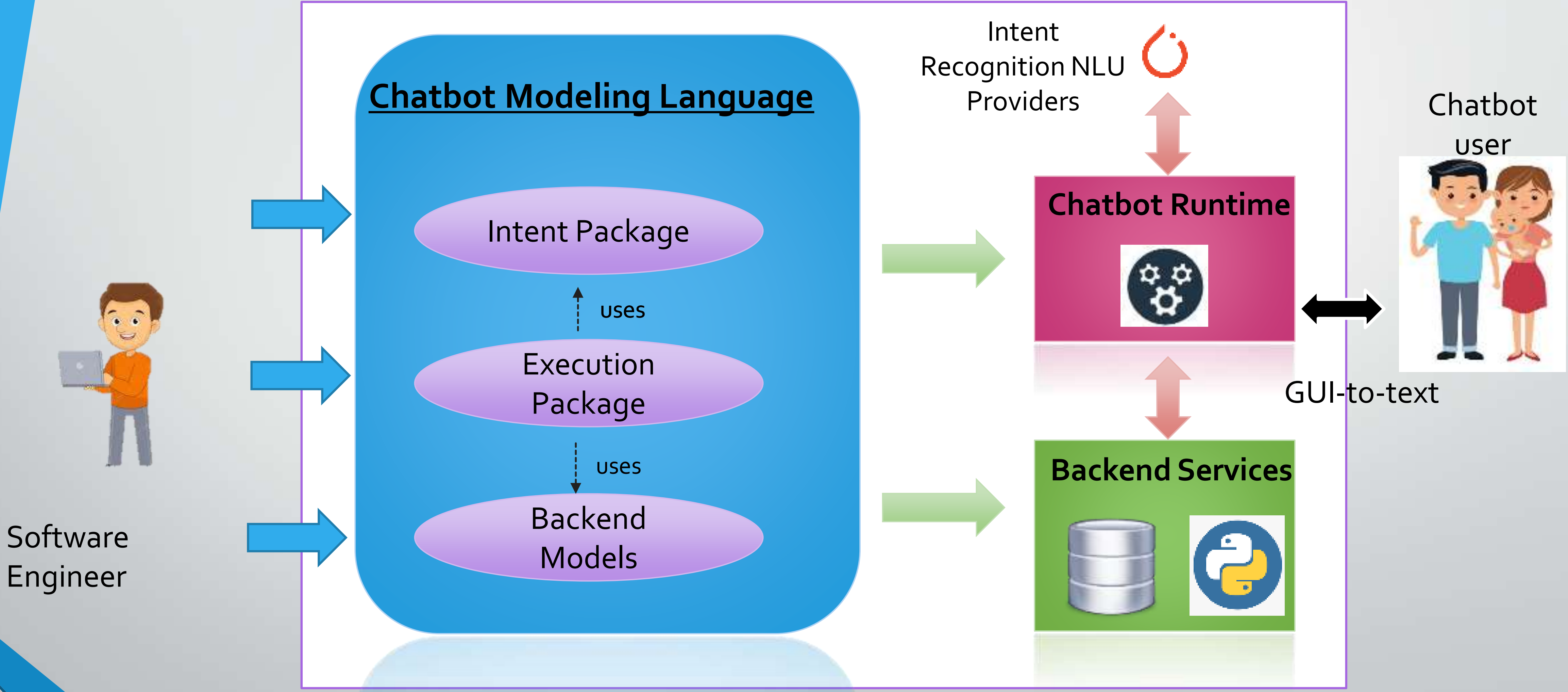
Image Processing and Recognition: How can you build an accurate image recognition system capable of identifying various types of skin rashes?

Database and Training Data: What kind of database and training data do you need to train the image recognition model effectively?

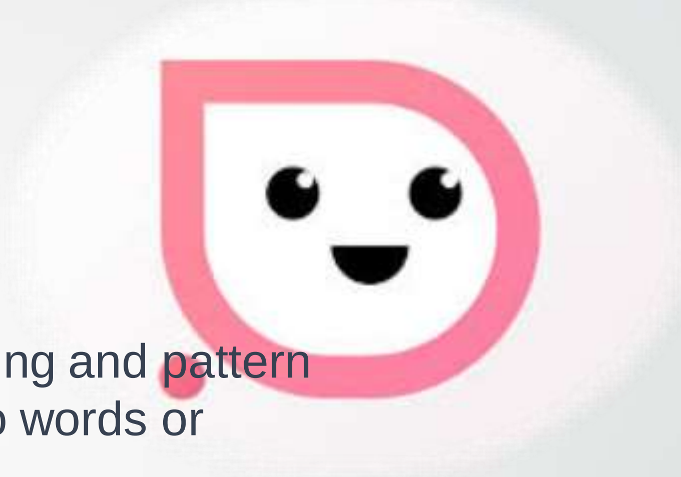
Accuracy and Reliability: How can you ensure the system's accuracy and reliability in identifying skin rash diseases?

Legal and Medical Regulations: Are there any legal and medical regulations that need to be considered when developing and deploying a system for disease identification?

Component Overview Diagram



Methodology – Chat-bot



1.Regular Expressions (regex): The **re** module is used for text processing and pattern matching using regular expressions. Here, it's used to split user input into words or tokens.

2.Natural Language Processing (NLP): NLP is a key technology in chatbot development. NLP techniques are used to understand and generate human-like text responses. Libraries like **NLTK (Natural Language Toolkit)** used for NLP tasks.

3. Response Generation: Based on predefined patterns and keywords provided in the **check_all_messages** function. It uses a **rule-based** approach, matching user input to predefined responses.

4. External Module (long_responses): An external module named **long_responses** which contains longer response messages. This module seems to contain detailed responses for various user queries. This demonstrates modularity separating responses from the main logic.

5. Interactive Loop: An interactive loop, continuously accepting user input and providing responses until the user types "exit."“

6. Chatbot Logic: The core logic of the chatbot involves matching user input to predefined keywords and selecting the response with the highest probability. It also checks for required words to ensure that certain conditions are met before providing a response.

7. Exit Command Handling: A mechanism to exit the loop when the user inputs "exit," allowing the user to quit the conversation.

1.Flask: Flask is a micro web framework for Python. It's used here to create a web server that handles HTTP requests and responses.

2.MongoDB: MongoDB is a NoSQL database used to store chat conversations. The **pymongo** library is used to connect to and interact with the MongoDB database.

1.HTML and Jinja2 Templates: HTML is used to create the structure of the web page, and Flask uses the Jinja2 template engine to dynamically generate HTML content and render it on the web page.

2.JSON (JavaScript Object Notation): JSON is used for data exchange between the web page and the Flask server. The **jsonify** function from Flask is used to send JSON responses to the client.

Training file

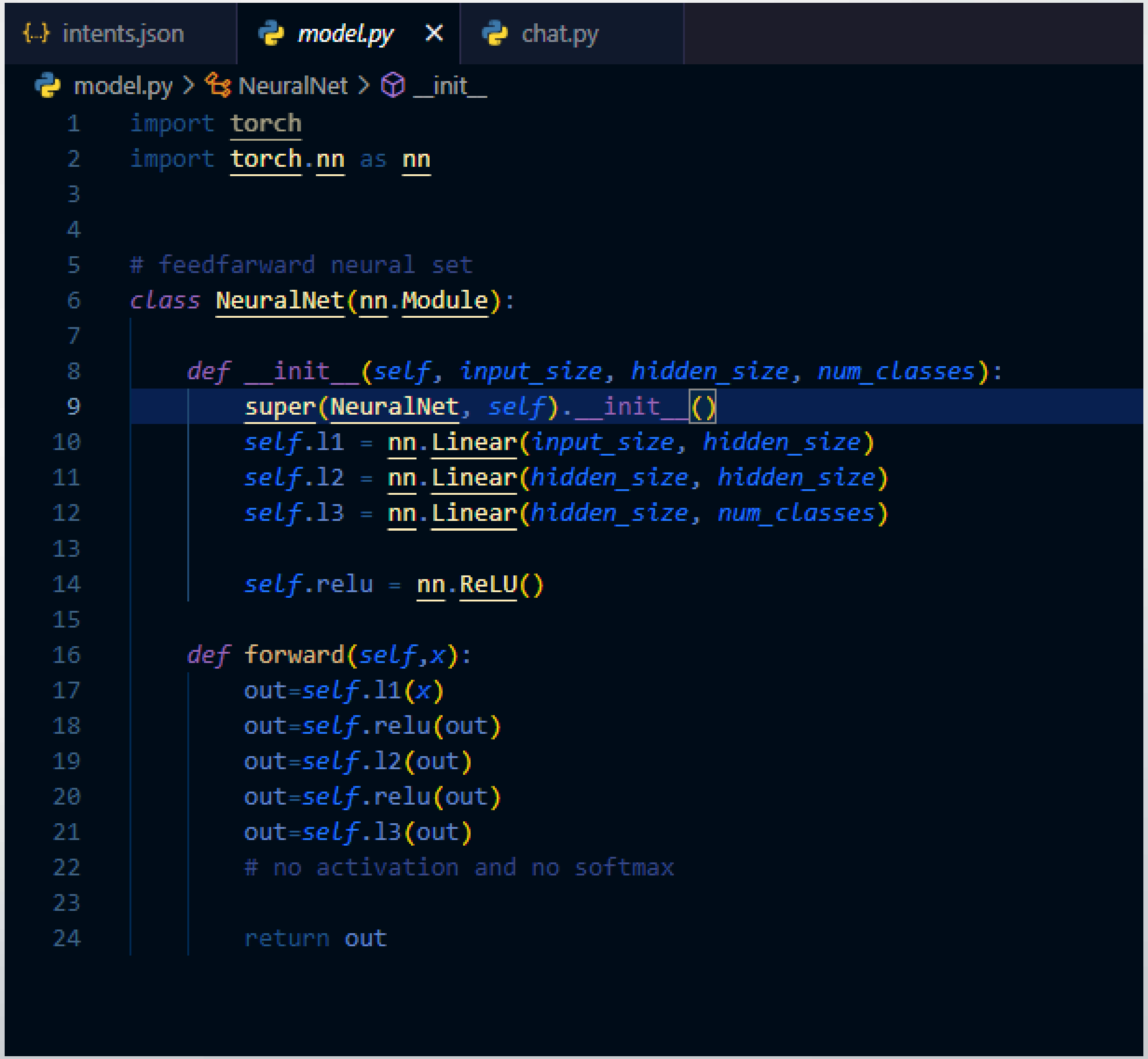
- The intent file,
- NLTK,
- Numpy,
- Pytorch,
- The dataset
- the model,

of the system are imported into the training file. The file is to train the model by the dataset by using each technique

- **numpy**: A library for numerical operations and array manipulation.
- **Pytorch:** open-source machine learning library that provides a flexible framework for building and training deep learning models efficiently.
- **Epoch**: refers to a single pass through the entire training dataset during the training process of a model. It helps optimize the model's parameters by updating them based on the calculated loss and the chosen optimization algorithm.

Model

Programming Language: The primary programming language used is **Python**. Python is a popular choice for building chatbots due to its simplicity and the availability of libraries and frameworks for natural language processing (NLP).



```python
import torch
import torch.nn as nn


# feedfarward neural set
class NeuralNet(nn.Module):

    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)

        self.relu = nn.ReLU()

    def forward(self,x):
        out=self.l1(x)
        out=self.relu(out)
        out=self.l2(out)
        out=self.relu(out)
        out=self.l3(out)
        # no activation and no softmax

        return out
```

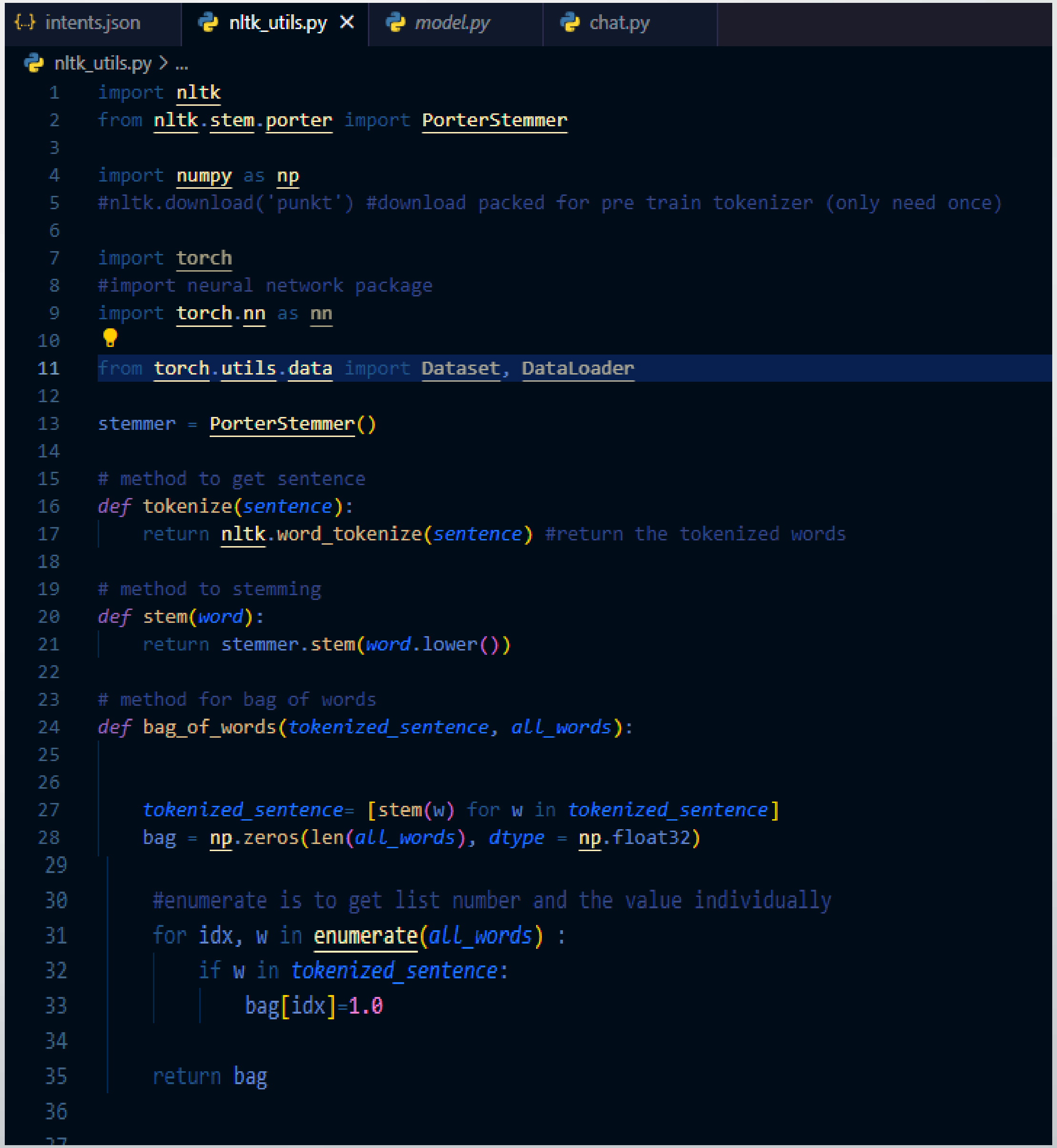
NLTK

NLTK stands for Natural Language Toolkit. This technique is used for tokenization, stemming, and getting the bag of words

Tokenization -Split strings into meaningful units

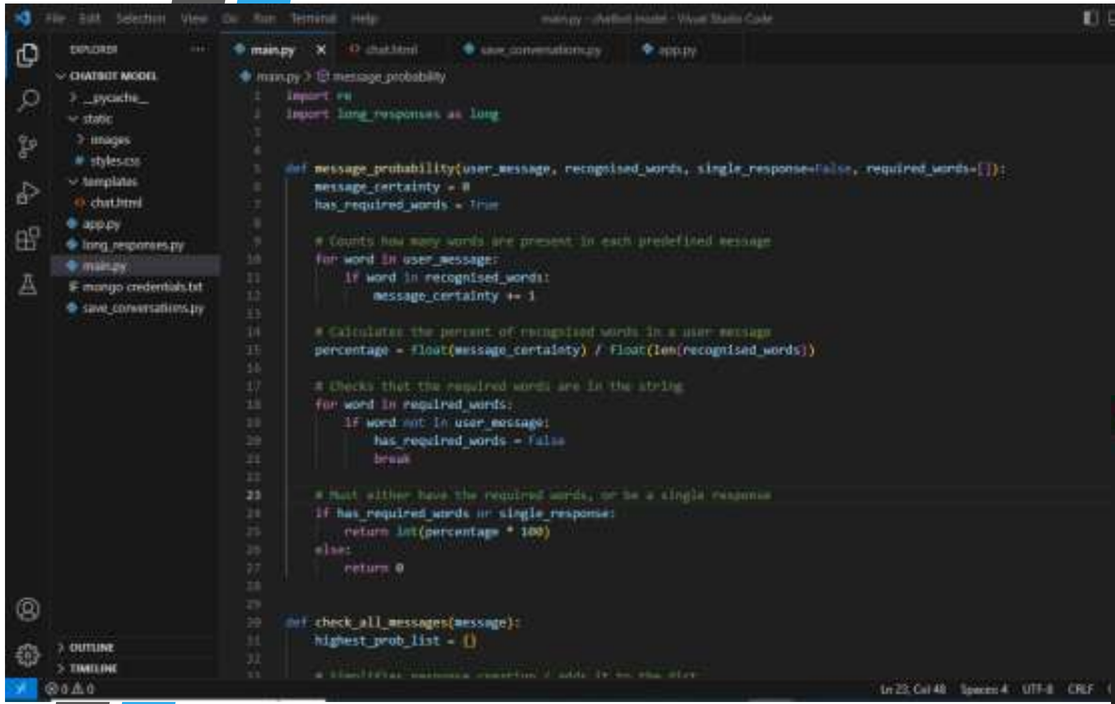
Stemming –Generate the root form by cutting the end off

Bag_of_words – the final outcome of binary values of stemmed words



```python
import nltk
from nltk.stem.porter import PorterStemmer

import numpy as np
#nltk.download('punkt') #download packed for pre train tokenizer (only need once)

import torch
#import neural network package
import torch.nn as nn

from torch.utils.data import Dataset, DataLoader

stemmer = PorterStemmer()

# method to get sentence
def tokenize(sentence):
    return nltk.word_tokenize(sentence) #return the tokenized words

# method to stemming
def stem(word):
    return stemmer.stem(word.lower())

# method for bag of words
def bag_of_words(tokenized_sentence, all_words):


    tokenized_sentence= [stem(w) for w in tokenized_sentence]
    bag = np.zeros(len(all_words), dtype = np.float32)

    #enumerate is to get list number and the value individually
    for idx, w in enumerate(all_words) :
        if w in tokenized_sentence:
            bag[idx]=1.0

    return bag

```

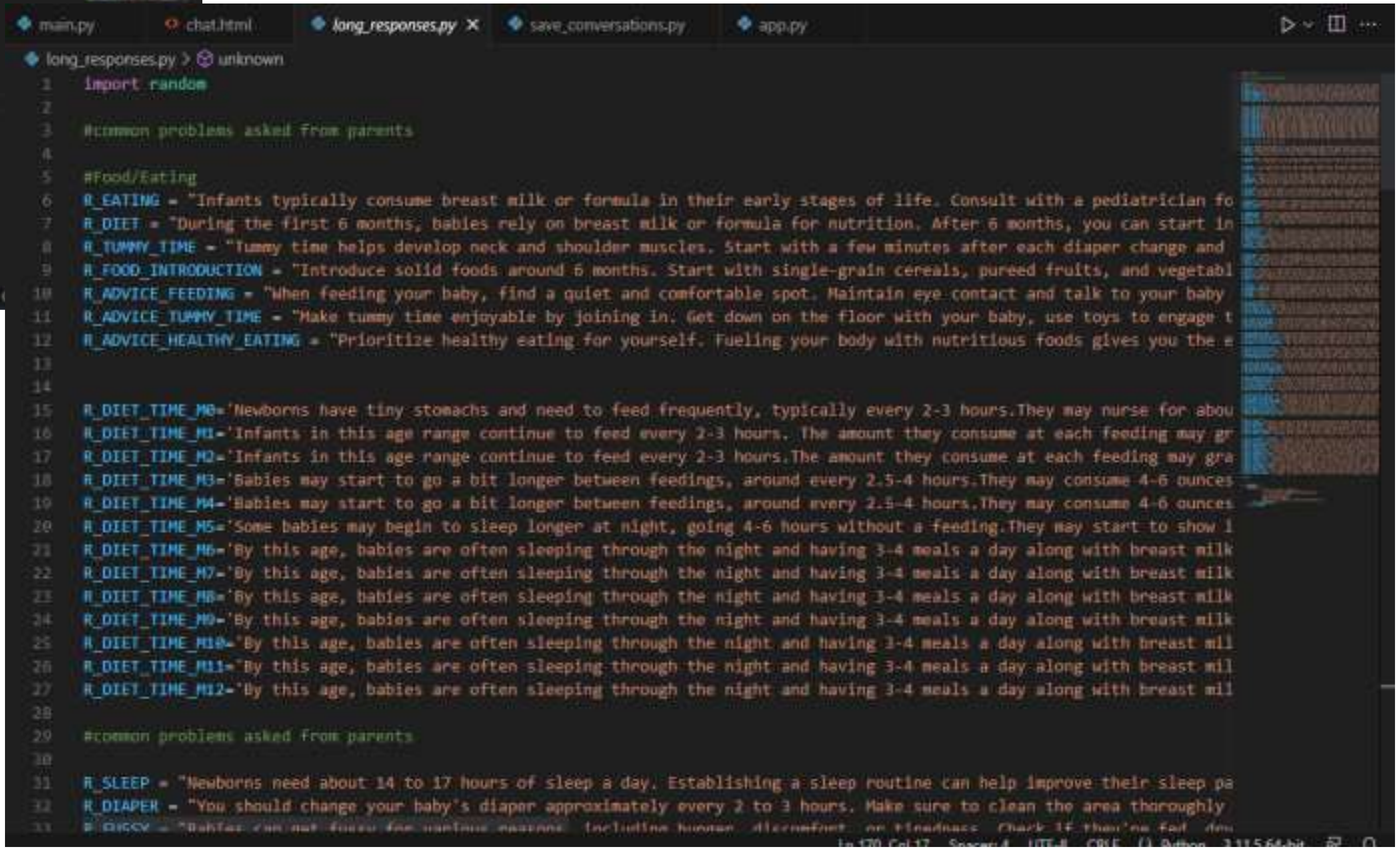
Developed solution – chat bot



```python
import re
import long_responses as long


def message_probability(user_message, recognised_words, single_response=False, required_words=[]):
    message_certainty = 0
    has_required_words = True

    # Counts how many words are present in each predefined message
    for word in user_message:
        if word in recognised_words:
            message_certainty += 1

    # Calculates the percent of recognised words in a user message
    percentage = float(message_certainty) / float(len(recognised_words))

    # Checks that the required words are in the string
    for word in required_words:
        if word not in user_message:
            has_required_words = False
            break

    # Must either have the required words, or be a single response
    if has_required_words or single_response:
        return int(percentage * 100)
    else:
        return 0


def check_all_messages(message):
    highest_prob_list = {}

    # Simplifies response creation / adds it to the dict
```

Model.py

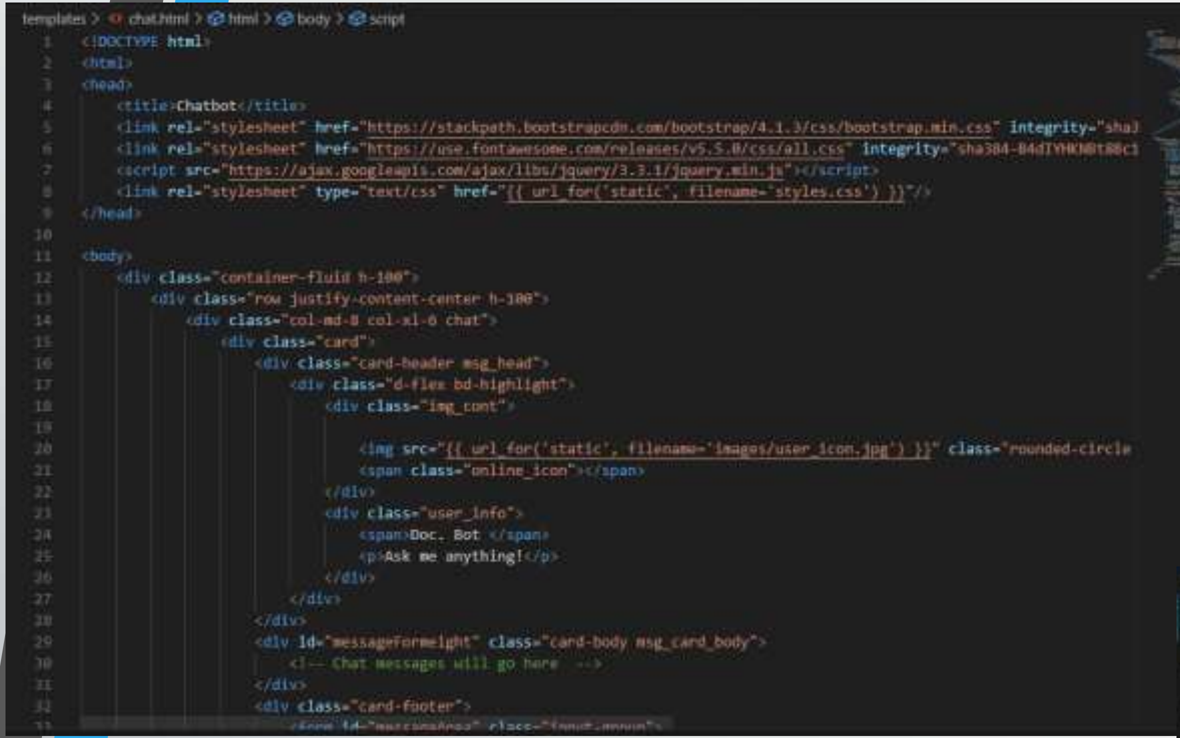
long_responses



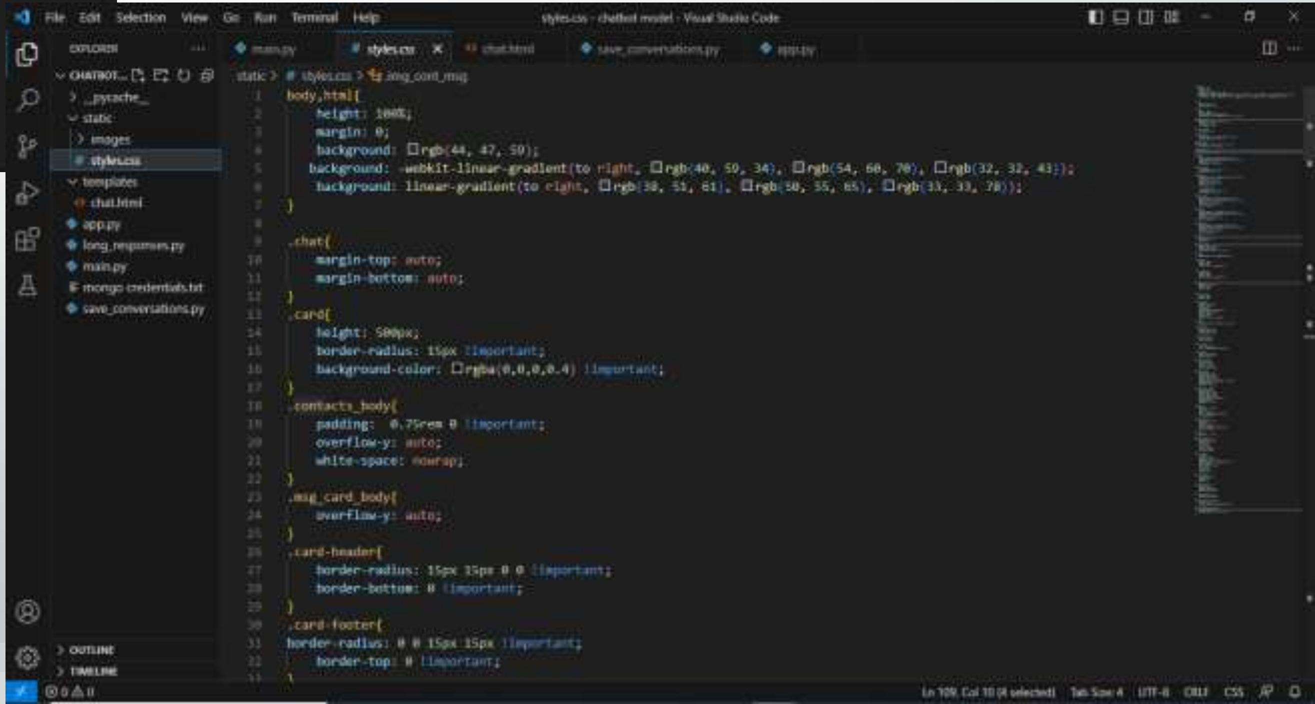
```python
import random

#common problems asked from parents

#Food/Eating
R_EATING = "Infants typically consume breast milk or formula in their early stages of life. Consult with a pediatrician fo
R_DIET = "During the first 6 months, babies rely on breast milk or formula for nutrition. After 6 months, you can start in
R_TUMMY_TIME = "Tummy time helps develop neck and shoulder muscles. Start with a few minutes after each diaper change and
R_FOOD_INTRODUCTION = "Introduce solid foods around 6 months. Start with single-grain cereals, pureed fruits, and vegetabl
R_ADVICE_FEEDING = "When feeding your baby, find a quiet and comfortable spot. Maintain eye contact and talk to your baby
R_ADVICE_TUMMY_TIME = "Make tummy time enjoyable by joining in. Get down on the floor with your baby, use toys to engage t
R_ADVICE_HEALTHY_EATING = "Prioritize healthy eating for yourself. Fueling your body with nutritious foods gives you the e


R_DIET_TIME_M0='Newborns have tiny stomachs and need to feed frequently, typically every 2-3 hours.They may nurse for abou
R_DIET_TIME_M1='Infants in this age range continue to feed every 2-3 hours. The amount they consume at each feeding may gr
R_DIET_TIME_M2='Infants in this age range continue to feed every 2-3 hours.The amount they consume at each feeding may gra
R_DIET_TIME_M3='Babies may start to go a bit longer between feedings, around every 2.5-4 hours.They may consume 4-6 ounces
R_DIET_TIME_M4='Babies may start to go a bit longer between feedings, around every 2.5-4 hours.They may consume 4-6 ounces
R_DIET_TIME_M5='Some babies may begin to sleep longer at night, going 4-6 hours without a feeding.They may start to show i
R_DIET_TIME_M6='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast milk
R_DIET_TIME_M7='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast milk
R_DIET_TIME_M8='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast milk
R_DIET_TIME_M9='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast milk
R_DIET_TIME_M10='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast mil
R_DIET_TIME_M11='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast mil
R_DIET_TIME_M12='By this age, babies are often sleeping through the night and having 3-4 meals a day along with breast mil

#common problems asked from parents

R_SLEEP = "Newborns need about 14 to 17 hours of sleep a day. Establishing a sleep routine can help improve their sleep pa
R_DIAPER = "You should change your baby's diaper approximately every 2 to 3 hours. Make sure to clean the area thoroughly
R_FUSSY = "Babies can get fussy for various reasons, including hunger, discomfort, or tiredness. Check if they're fed, dry
```

Chat file

This code uses by the user interaction HTML, CSS, JavaScript



```html
<!DOCTYPE html>
<html>
<head>
    <title>Chatbot</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha3
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc1
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}"/>
</head>

<body>
    <div class="container-fluid h-100">
        <div class="row justify-content-center h-100">
            <div class="col-md-8 col-xl-6 chat">
                <div class="card">
                    <div class="card-header msg_head">
                        <div class="d-flex bd-highlight">
                            <div class="img_cont">

                                <img src="{{ url_for('static', filename='images/user_icon.jpg') }}" class="rounded-circle
                                <span class="online_icon"></span>
                            </div>
                            <div class="user_info">
                                <span>Doc. Bot </span>
                                <p>Ask me anything!</p>
                            </div>
                        </div>
                    </div>
                    <div id="messageFormeight" class="card-body msg_card_body">
                        <!-- Chat messages will go here -->
                    </div>
                    <div class="card-footer">
                        <form id="messageArea" class="input-group">
```

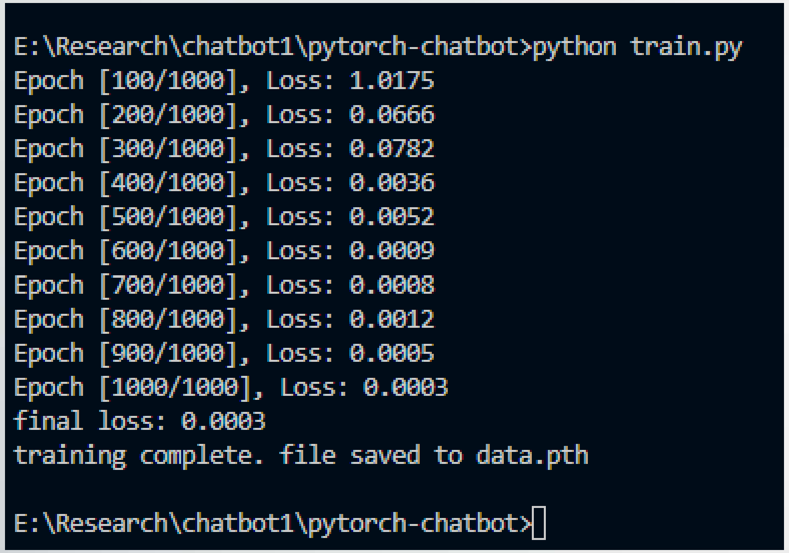


```css
body,html{
    height: 100%;
    margin: 0;
    background: rgb(44, 47, 59);
   background: -webkit-linear-gradient(to right, rgb(40, 59, 34), rgb(54, 60, 70), rgb(32, 32, 43));
    background: linear-gradient(to right, rgb(38, 51, 61), rgb(50, 55, 65), rgb(33, 33, 78));
}

.chat{
    margin-top: auto;
    margin-bottom: auto;
}
.card{
    height: 500px;
    border-radius: 15px !important;
    background-color: rgba(0,0,0,0.4) !important;
}
.contacts_body{
    padding:  0.75rem 0 !important;
    overflow-y: auto;
    white-space: nowrap;
}
.msg_card_body{
    overflow-y: auto;
}
.card-header{
    border-radius: 15px 15px 0 0 !important;
    border-bottom: 0 !important;
}
.card-footer{
border-radius: 0 0 15px 15px !important;
    border-top: 0 !important;
}
```

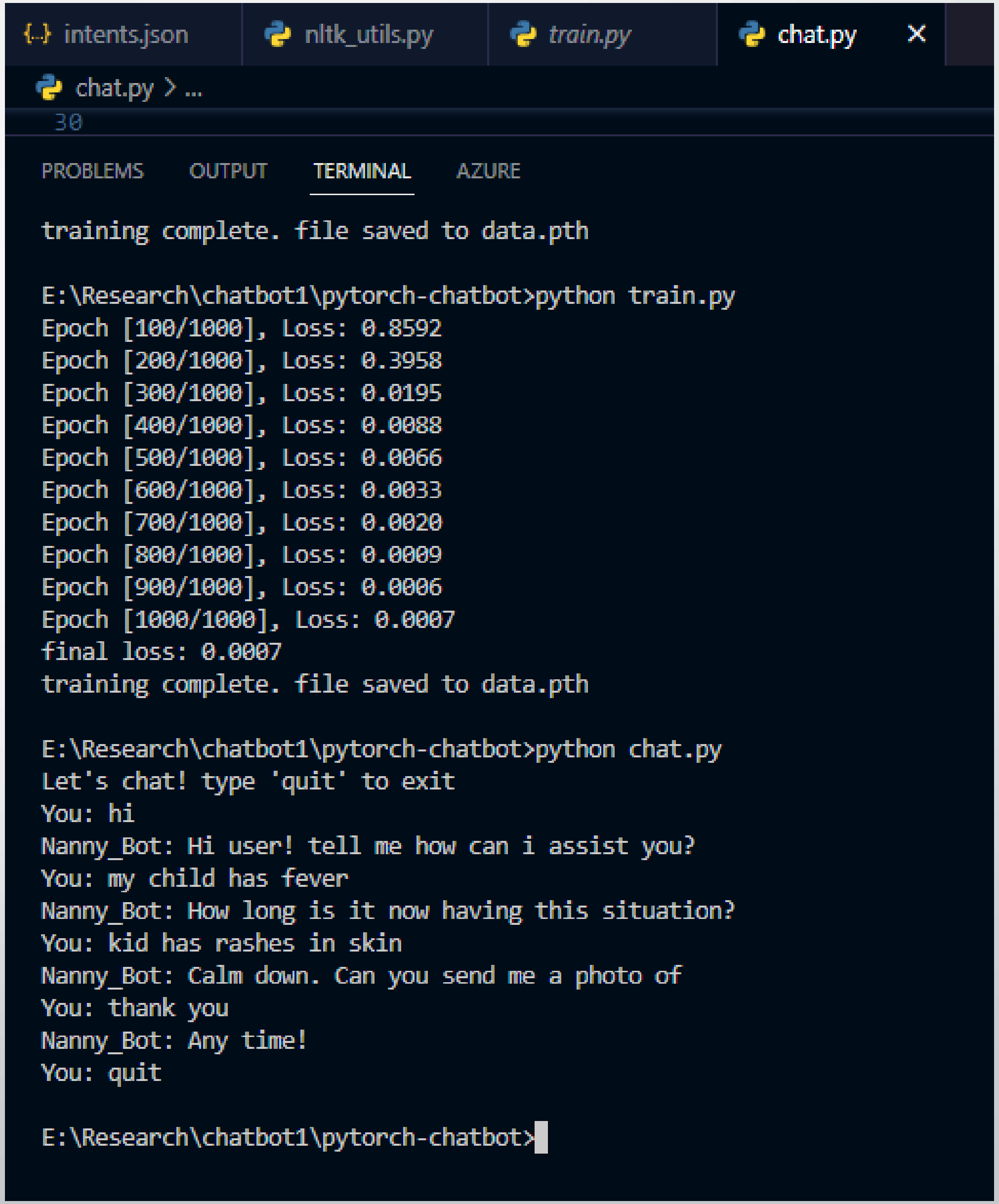
SLIIT FACULTY OF COMPUTING

Final output

- Training result



```
E:\Research\chatbot1\pytorch-chatbot>python train.py
Epoch [100/1000], Loss: 1.0175
Epoch [200/1000], Loss: 0.0666
Epoch [300/1000], Loss: 0.0782
Epoch [400/1000], Loss: 0.0036
Epoch [500/1000], Loss: 0.0052
Epoch [600/1000], Loss: 0.0009
Epoch [700/1000], Loss: 0.0008
Epoch [800/1000], Loss: 0.0012
Epoch [900/1000], Loss: 0.0005
Epoch [1000/1000], Loss: 0.0003
final loss: 0.0003
training complete. file saved to data.pth

E:\Research\chatbot1\pytorch-chatbot>
```



```
training complete. file saved to data.pth

E:\Research\chatbot1\pytorch-chatbot>python train.py
Epoch [100/1000], Loss: 0.8592
Epoch [200/1000], Loss: 0.3958
Epoch [300/1000], Loss: 0.0195
Epoch [400/1000], Loss: 0.0088
Epoch [500/1000], Loss: 0.0066
Epoch [600/1000], Loss: 0.0033
Epoch [700/1000], Loss: 0.0020
Epoch [800/1000], Loss: 0.0009
Epoch [900/1000], Loss: 0.0006
Epoch [1000/1000], Loss: 0.0007
final loss: 0.0007
training complete. file saved to data.pth

E:\Research\chatbot1\pytorch-chatbot>python chat.py
Let's chat! type 'quit' to exit
You: hi
Nanny_Bot: Hi user! tell me how can i assist you?
You: my child has fever
Nanny_Bot: How long is it now having this situation?
You: kid has rashes in skin
Nanny_Bot: Calm down. Can you send me a photo of
You: thank you
Nanny_Bot: Any time!
You: quit

E:\Research\chatbot1\pytorch-chatbot>
```

Methodology – Skin Rashes Prediction

1.**Flask**: Flask is a micro web framework for Python. It's used to create a web server that handles HTTP requests and responses.

2.**NumPy**: NumPy is a Python library for numerical computations. It's used for mathematical operations, especially in the context of machine learning.

3.**TensorFlow and Keras**: TensorFlow is an open-source machine learning framework, and Keras is an API that runs on top of TensorFlow. The code uses Keras to load a pre-trained skin rash classification model and make predictions based on uploaded images.

4.**Werkzeug**: Werkzeug is a utility library for Python web applications. It's used here for secure file uploads.

5.**HTML Templates**: The code includes HTML templates to render web pages for uploading images and displaying results. It uses the Jinja2 template engine, which is integrated with Flask.

6.**Static Files**: Static files, such as CSS stylesheets and images, are served using Flask's **static_url_path** configuration.

7.**File Uploads**: Flask handles file uploads from users, allowing them to upload skin rash images for classification.

8.Model Loading: The code defines a function to load a pre-trained skin rash classification model stored in an H5 file. The model is loaded using Keras.

9.Image Preprocessing: Images uploaded by users are preprocessed before being fed into the model. The preprocessing includes resizing the image to a specific target size and normalizing pixel values

10.Model Prediction: The code uses the loaded model to make predictions on uploaded images. It assumes that the model outputs probabilities for different skin rash types and selects the class with the highest probability as the predicted class.

11.Allowed File Extensions: It checks if the uploaded file has a valid image file extension (e.g., .jpg, .jpeg, .png) to ensure that only valid image files are processed.

12.Error Handling: The code handles various error cases, such as when no file is selected or when an invalid file format is uploaded. It provides error messages to the user.

13.Rendering Results: After making predictions, the code renders a results page that displays the uploaded image and the predicted skin rash type.

14.Running the App: The code includes a block to run the Flask application when the script is executed.

Training file

1.TensorFlow and Keras

2.NumPy

3.Matplotlib

4.Pandas

5.ImageDataGenerator

6.Google Colaboratory

7.File Management

8.Convolutional Neural Networks (CNNs)

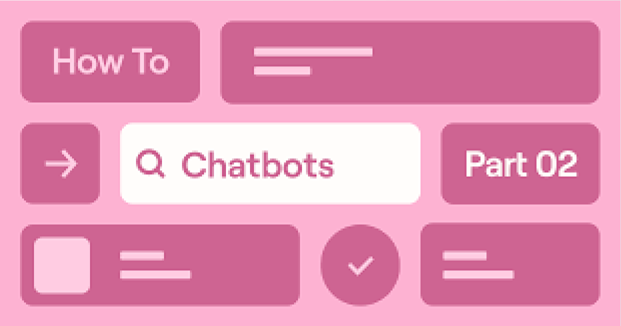
9.Data Augmentation

10.One-Hot Encoding

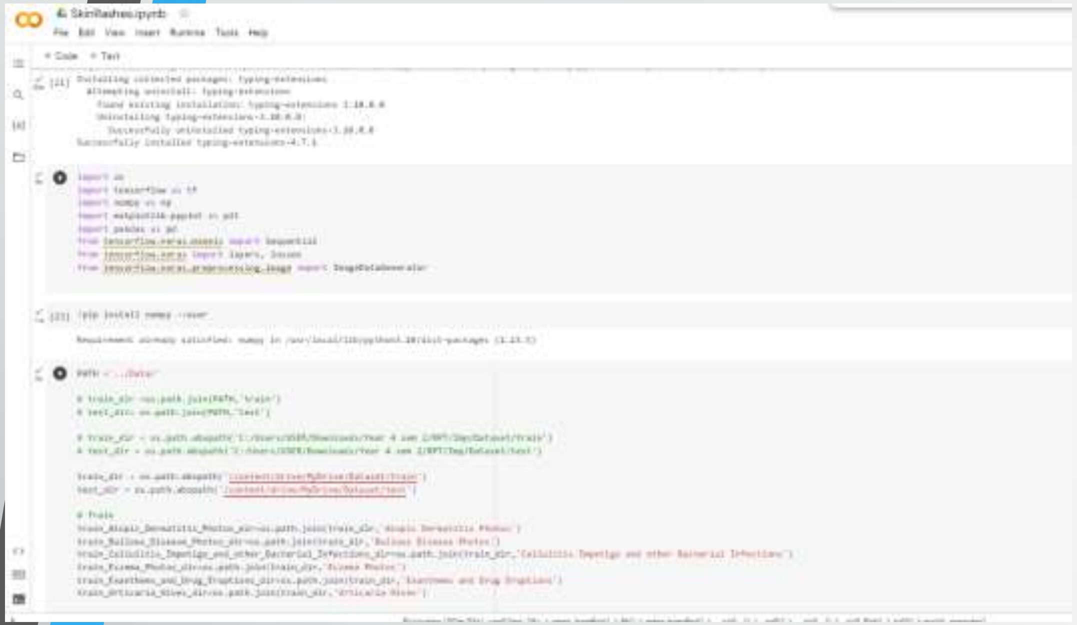
11.Plotting

12.Google Drive Integration

13.Model Saving



Developed solution – skin rashes prediction

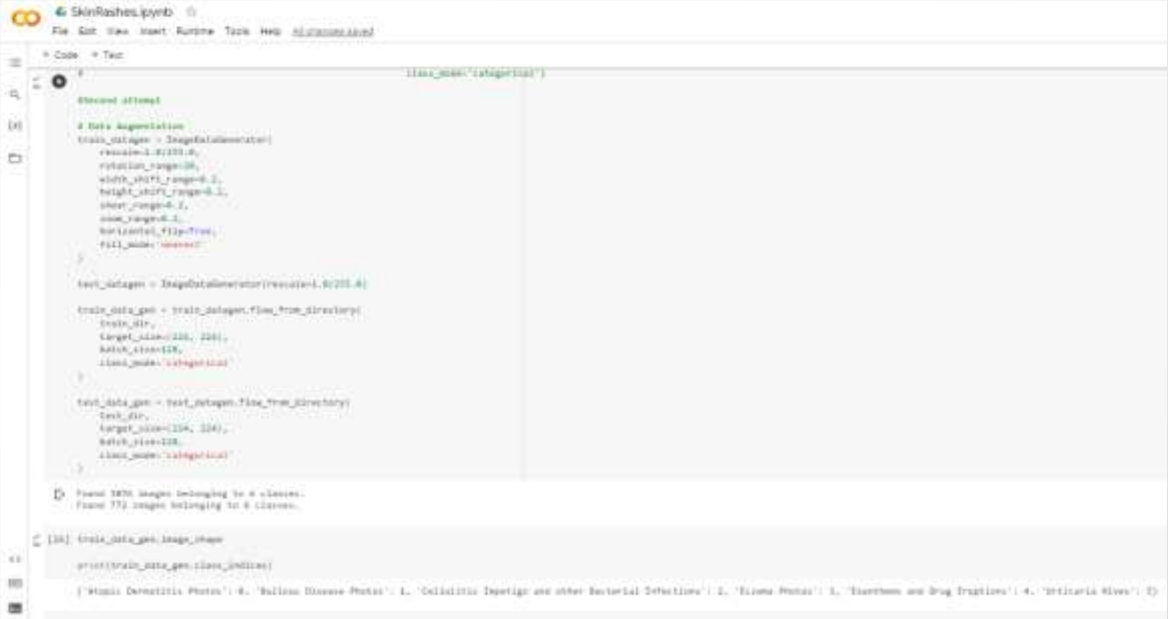


```
[21] Installing collected packages: typing-extensions
       Attempting uninstall: typing-extensions
         Found existing installation: typing-extensions 3.10.0.0
         Uninstalling typing-extensions-3.10.0.0:
           Successfully uninstalled typing-extensions-3.10.0.0
     Successfully installed typing-extensions-4.7.1
```

```
import os
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers, losses
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[23] !pip install numpy --user

     Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.23.5)
```

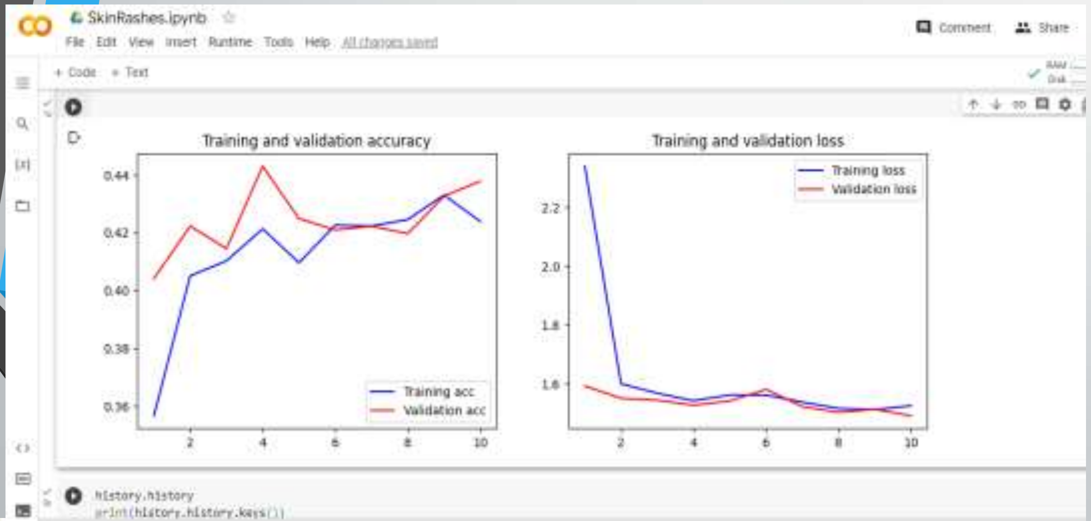
```
PATH = '../Data'

# train_dir =os.path.join(PATH,'train')
# test_dir= os.path.join(PATH,'test')

# train_dir = os.path.abspath('C:/Users/USER/Downloads/Year 4 sem 2/RP/Tmp/Dataset/train')
# test_dir = os.path.abspath('C:/Users/USER/Downloads/Year 4 sem 2/RP/Tmp/Dataset/test')

train_dir = os.path.abspath('/content/drive/MyDrive/Dataset/train')
test_dir = os.path.abspath('/content/drive/MyDrive/Dataset/test')

# Train
train_Atopic_Dermatitis_Photos_dir=os.path.join(train_dir,'Atopic Dermatitis Photos')
train_Bullous_Disease_Photos_dir=os.path.join(train_dir,'Bullous Disease Photos')
train_Cellulitis_Impetigo_and_other_Bacterial_Infections_dir=os.path.join(train_dir,'Cellulitis Impetigo and other Bacterial Infections')
train_Eczema_Photos_dir=os.path.join(train_dir,'Eczema Photos')
train_Exanthems_and_Drug_Eruptions_dir=os.path.join(train_dir,'Exanthems and Drug Eruptions')
train_Urticaria_Hives_dir=os.path.join(train_dir,'Urticaria Hives')
```



```
#                                              class_mode='categorical')

#Second attempt

# Data Augmentation
train_datagen = ImageDataGenerator(
    rescale=1.0/255.0,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

test_datagen = ImageDataGenerator(rescale=1.0/255.0)

train_data_gen = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=128,
    class_mode='categorical'
)

test_data_gen = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=128,
    class_mode='categorical'
)
```

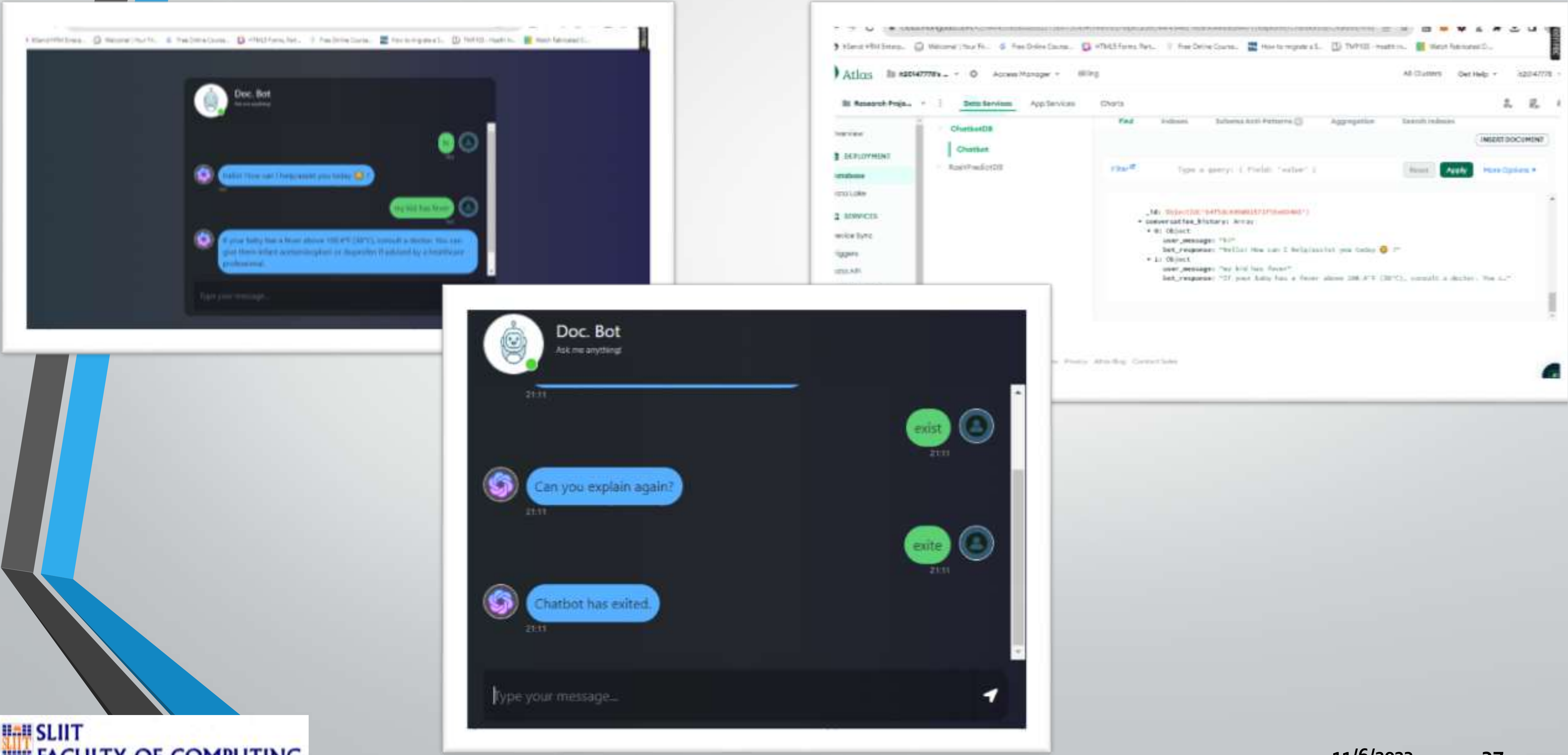
```
Found 3870 images belonging to 6 classes.
Found 773 images belonging to 6 classes.
```

```
[38] train_data_gen.image_shape

     print(train_data_gen.class_indices)

     {'Atopic Dermatitis Photos': 0, 'Bullous Disease Photos': 1, 'Cellulitis Impetigo and other Bacterial Infections': 2, 'Eczema Photos': 3, 'Exanthems and Drug Eruptions': 4, 'Urticaria Hives': 5}
```



```
history.history
print(history.history.keys())
```

Results and Discussion – Chat bot



Doc. Bot

hi

Hello! How can I help/assist you today 😊 ?

my kid has fever

If your baby has a fever above 100.4°F (38°C), consult a doctor. You can give them infant acetaminophen or ibuprofen if advised by a healthcare professional.

Type your message...

```
_id: ObjectId('[illegible]')
▾ conversation_history: Array
  ▾ 0: Object
      user_message: "hi"
      bot_response: "Hello! How can I help/assist you today 😊 ?"
  ▾ 1: Object
      user_message: "my kid has fever"
      bot_response: "If your baby has a fever above 100.4°F (38°C), consult a doctor. You c…"
```

Doc. Bot

Ask me anything!

21:11

exist

21:11

Can you explain again?

21:11

exite

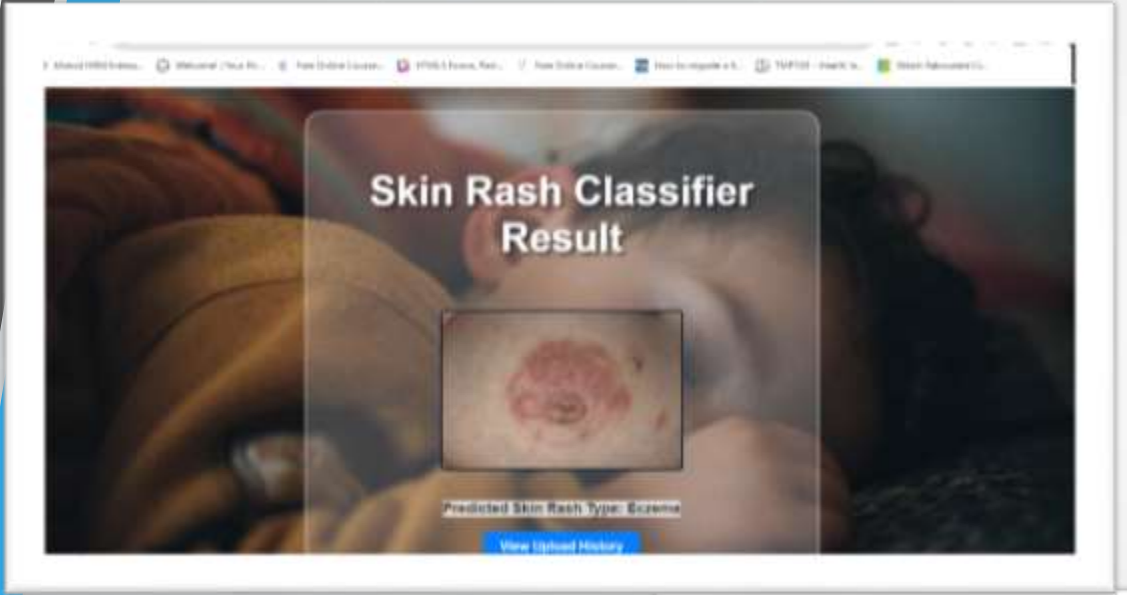
21:11

Chatbot has exited.

21:11

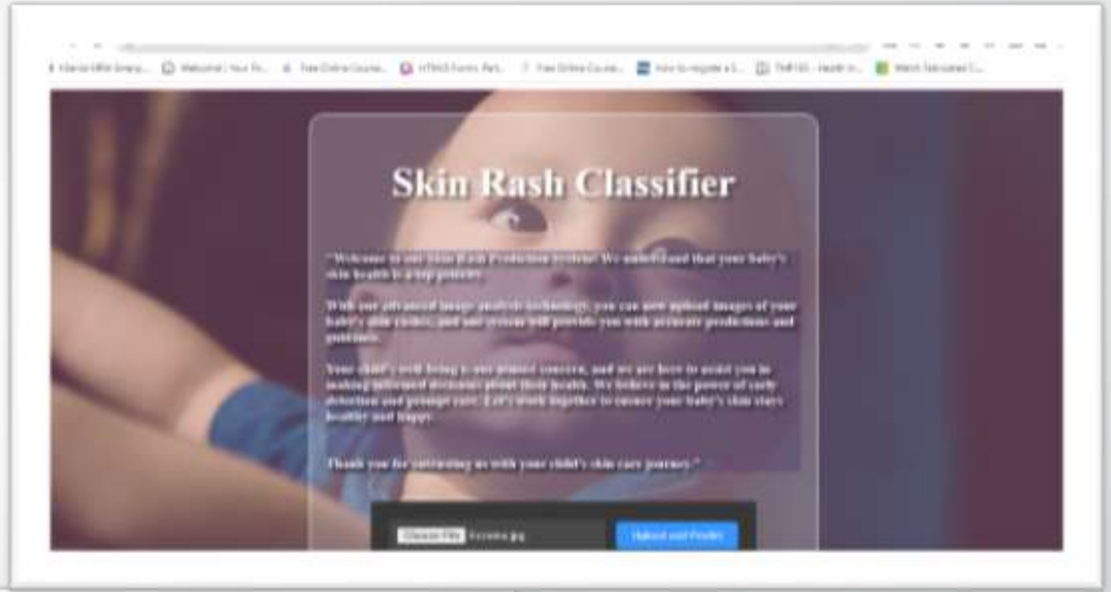
Type your message...

Results and Discussion – Skin Rashes Prediction



Skin Rash Classifier Result

Predicted Skin Rash Type: Eczema



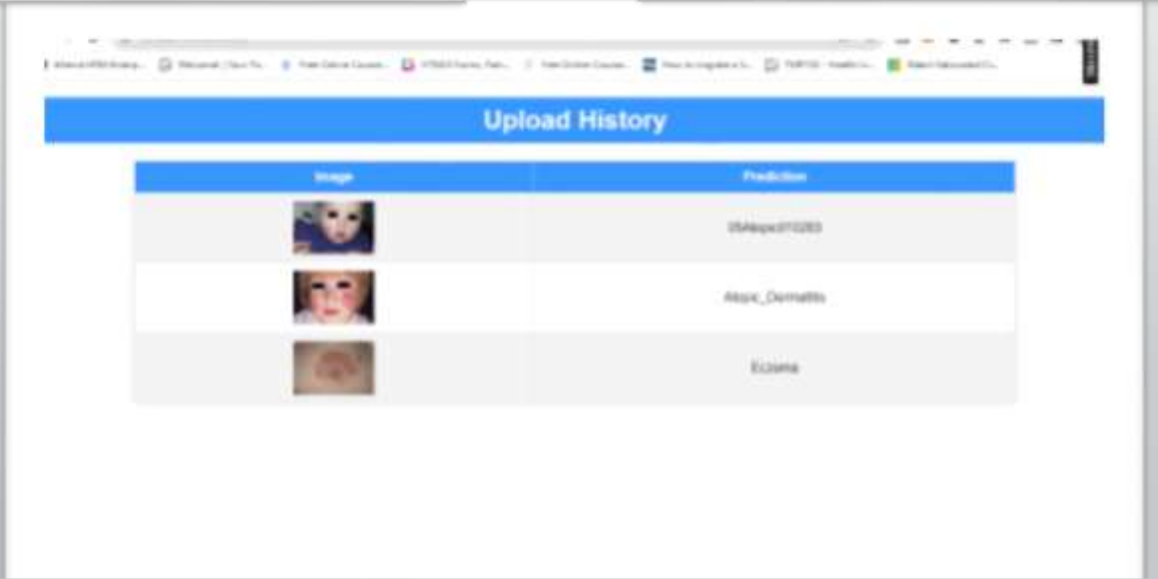
Skin Rash Classifier

"Welcome to our Skin Rash Prediction system! We understand that your baby's skin health is a top priority.

With our advanced image analysis technology, you can now upload images of your baby's skin rashes, and our system will provide you with accurate predictions and guidance.

Your child's well-being is our utmost concern, and we are here to assist you in making informed decisions about their health. We believe in the power of early detection and prompt care. Let's work together to ensure your baby's skin stays healthy and happy.

Thank you for entrusting us with your child's skin care journey."



Upload History

Image	Prediction
	05Atopic010283
	Atopic_Dermatitis
	Eczema

IT20405090 | Vanhoff R. L.

Growth Level Prediction

Software Engineering

Background

- After the proposal presentation the component was asked to change to a growth level predictor instead of a nutritional level identification and the functionality was changed from image processing to training a model to predict the growth level, by the panel.
- Information was gathered from the specialists and doctors in order to gather information for the predictions to make it more accurate and to discuss on how the functionality can be done.
- But after the process, it was stated that the growth level cannot be measured without the height, weight and age-appropriate other measures with just the behavioral milestones.
- As the final decision, the component was changed to a prediction function where according to the given measures of a child, the growth level can be predicted by the model.

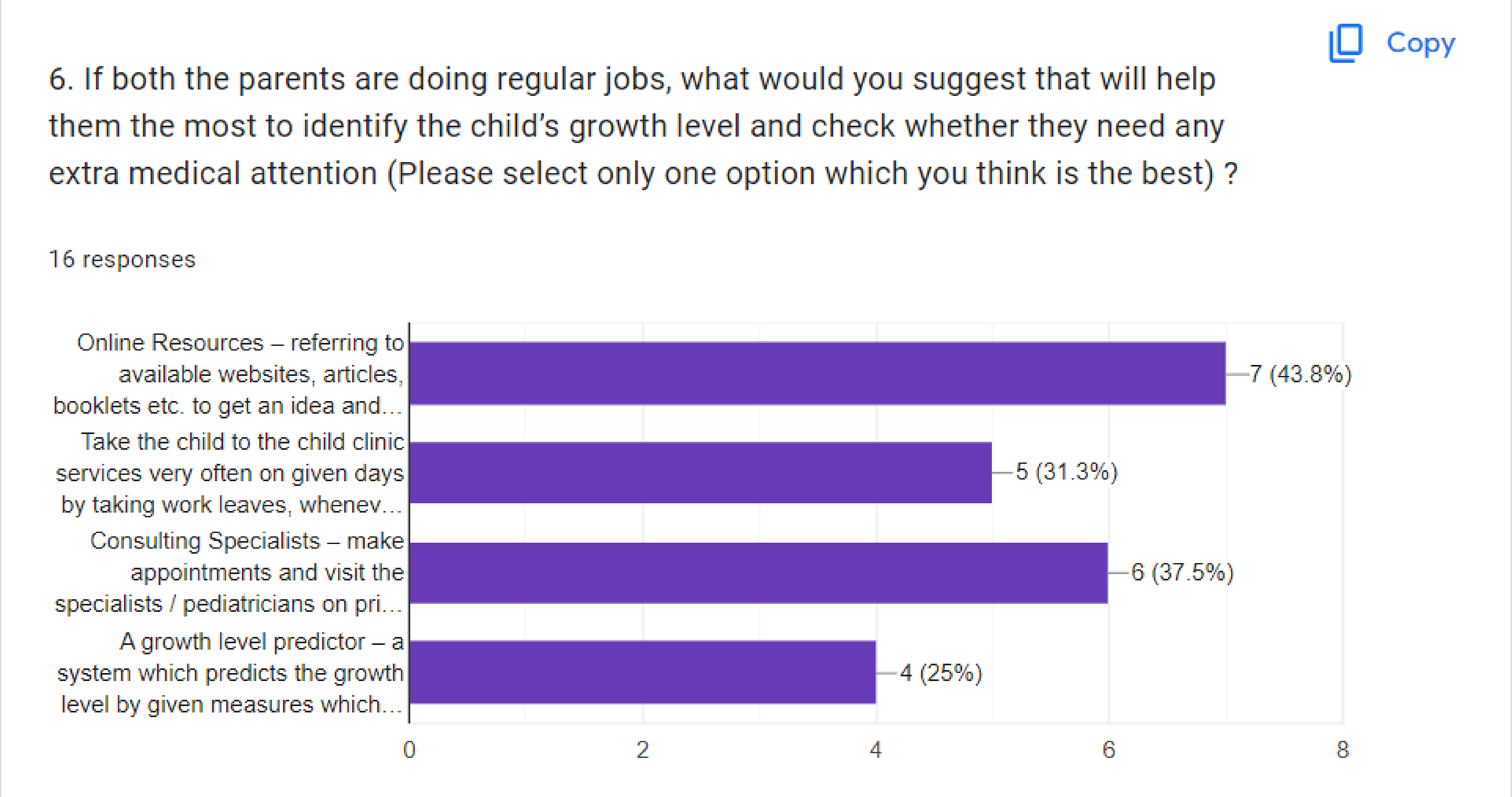
Information Gathering

- Several interviews were held with the doctors to discuss about the changed function before finalizing the component.

✓ **Dr. Udara Ariyasinghe - Arogya Family Medical Center**

✓ **Dr. Thilini Amarathunga - Muttur Hospital**

- Investigations were done by exploring the available systems, past research papers, tutorials and other related materials to get a clear understanding about doing predictions using a model.
- Referred online materials such as YouTube videos, online surveys, WHO documentation and other medicine related documentation to use the correct information to train the model and start the implementations.



6. If both the parents are doing regular jobs, what would you suggest that will help them the most to identify the child's growth level and check whether they need any extra medical attention (Please select only one option which you think is the best) ?

16 responses

Component Discussion

The Function's Primary Goal - The purpose is to give parents an automated method for determining and tracking their infants' growth levels. The function estimates the developmental stage based on several factors about a kid, including weight, height, age, and other development milestones.

The following are the function's main objectives:

- **Development Level Prediction**: The Decision Tree Regressor method, a machine learning approach, is used by the function to forecast newborns' development levels. The function trains a regression model to precisely forecast the growth level based on the input measurements by examining the provided dataset, which comprises development milestones data for children aged 0–5 years.
- **Precautionary Measures:** If the expected growth level suggests overgrowth or undergrowth, the function is intended to help parents take the appropriate safety measures. Parents can better comprehend their baby's development and seek the necessary medical advice or intervention by having insights about the child's growth trajectory.
- **User-Friendly Interfaces**: The function is incorporated into a user-friendly front-end web application that is a full-stack web application. Parents may easily input their baby's measurements into the interface to get a prediction of growth level. To make it simple for parents to understand and respond to the results, the interface also displays the forecast as a % along with corresponding categories of poor, fair, good, and excellent.

By emphasizing these crucial elements, the function intends to give parents a useful and instructive tool for tracking their child's development and encouraging proactive healthcare management.

Component Discussion cont.

User Workflow

- **Enter Child's Information**: Provide the child's age, height, weight, and select or input information about their developmental milestones in various domains, such as motor skills, problem-solving, communication, and emotional development.
- **Predict Growth Level**: Click the "Predict the Growth" button to initiate the prediction process. The system will process the entered data and calculate the child's growth level.
- **View Result**: Once the prediction is complete, view the predicted growth level displayed on the screen. Take note of the result and any recommendations provided by the application.

Impact and Future Scope

Impact:

- Empowers parents with valuable insights into their child's progress.
- Enables early detection of developmental delays or potential issues.
- Facilitates timely interventions and support.
- Promotes proactive parenting by offering personalized recommendations.
- Contributes to informed decision-making for child development.

Future Scope:

- Calculation of different milestones and personalized recommendations.
- Expansion of assessed milestones and incorporation of machine learning techniques.
- Integration of a comprehensive growth database for accurate comparisons.
- Inclusion of educational resources and interactive features.
- Continuous evolution and advancements as a valuable parenting tool.

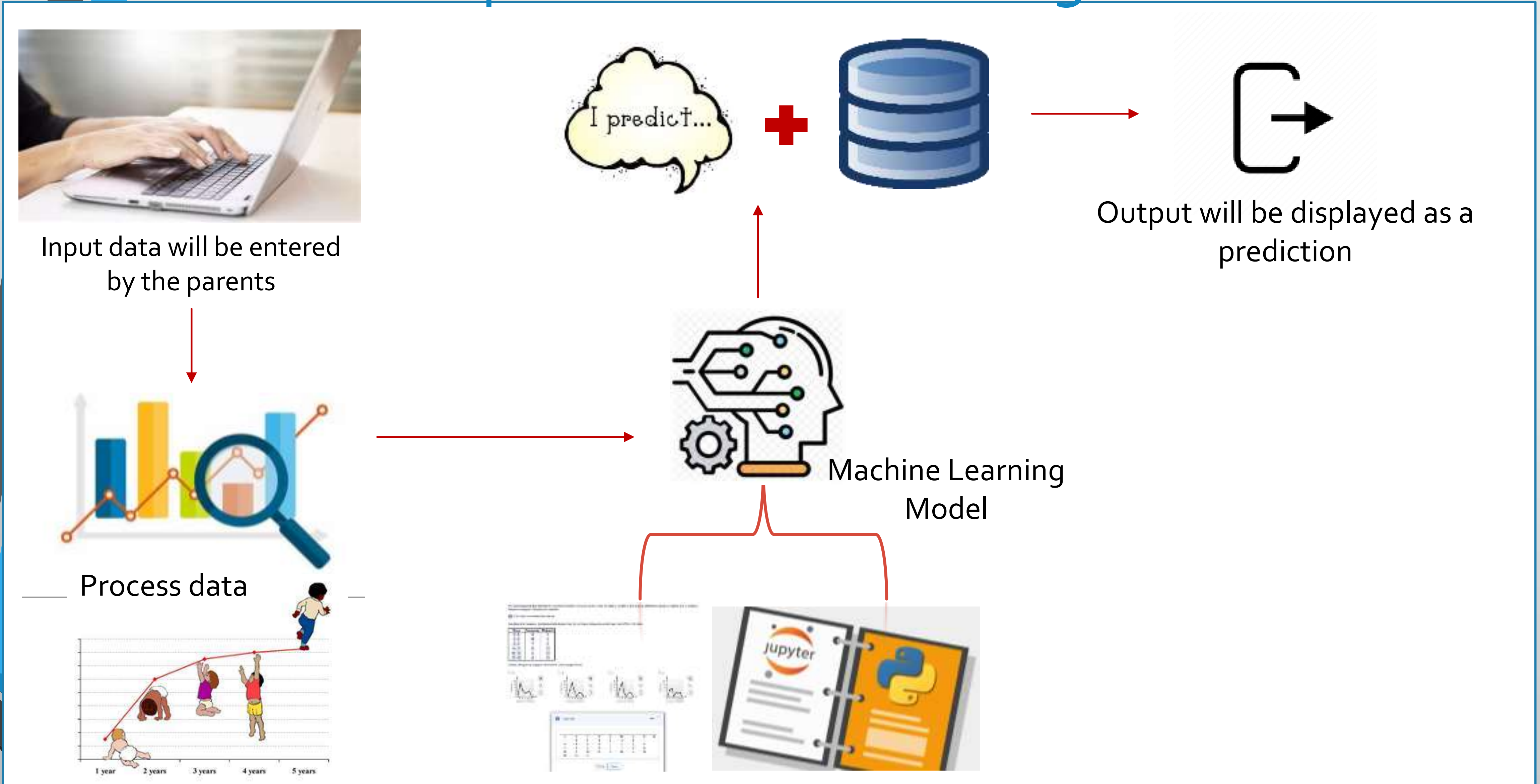
System Workflow

- **Data collection**: The system begins by gathering a dataset that includes information on the developmental stages of children aged 0 to 5. The growth prediction model is trained using the dataset as its foundation.
- **Data preprocessing**: To extract pertinent characteristics, the collected dataset is preprocessed. The algorithm in this instance chooses particular metrics like weight, height, age, emotional development, social skills, emotional issues, and growth rate. In addition to addressing missing values, categorical variables must be encoded using LabelEncoder, and the data must be split into training and testing sets using train_test_split.
- **Model Training**: The system trains the growth predictor model using the DecisionTreeRegressor method from the scikit-learn package. The selected characteristics and accompanying growth levels are included in the training dataset, which is used to train the model.
- **Model Evaluation**: Using the testing dataset, the model is assessed after training. To evaluate the model's success, the system predicts the growth levels for the testing data and contrasts them with the actual growth levels. The accuracy_score measure is used to determine the model's accuracy.

System Workflow cont.

- **Model Persistence**: After the model has been trained and assessed, it is stored using the pickle library in a serialized manner. The model is saved in a file along with the label encoders required to encode categorical variables.
- **Front-End Integration**: A full-stack online application incorporates the growth prediction model. Parents can pick certain observations or activities pertaining to their baby's growth and development using checkboxes on the front-end interface.
- **Prediction Calculation**: The system determines the score based on the number of selected actions when the user checks the appropriate checkboxes. The ratio of the number of chosen activities to the total number of potential actions is multiplied by 100 to get the score, which is then expressed as a percentage.
- **Forecast Interpretation**: Based on the estimated %, the algorithm chooses one of four growth forecast categories (poor, fair, good, or great). For example, 80% for exceptional, 60% for good, and 40% for fair, are used as specific thresholds to determine which category a forecast belongs in.
- **Results Presentation**: The system shows the user the growth projection category. Based on the chosen observations, the category represents the baby's growth stage. On the front-end interface, the results are displayed in an approachable manner, giving parents useful information about their child's development and the essential safety measures to be implemented.

Component Overview Diagram



Input data will be entered by the parents

Process data

Machine Learning Model

Output will be displayed as a prediction

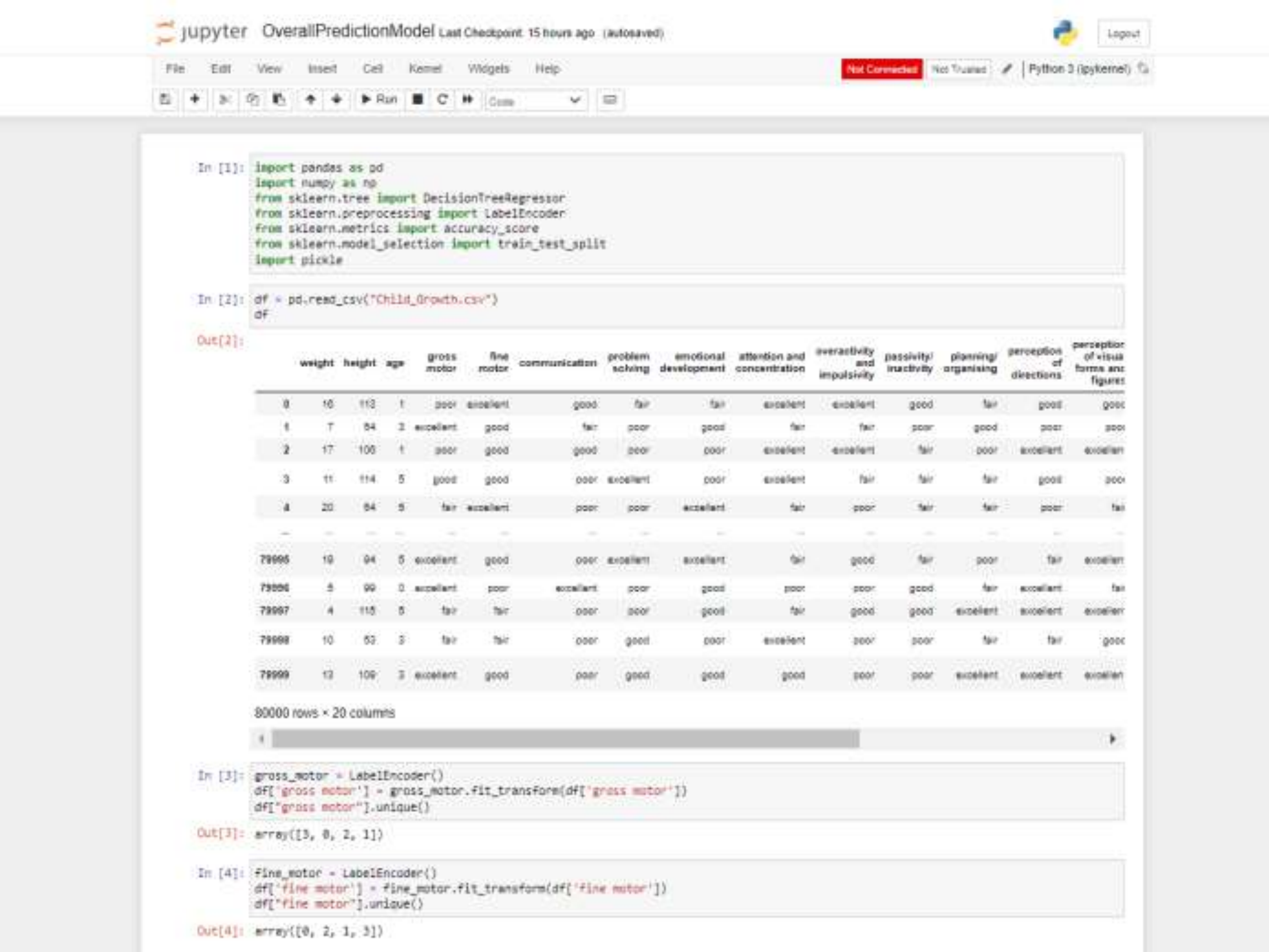
Train the Machine Learning model using a dataset

Implementation

Developing and training a model

Libraries:

- **numpy**: A library for numerical operations and array manipulation.
- **pandas**: A library for data manipulation and analysis.
- **sklearn.tree.DecisionTreeRegressor**: A class from scikit-learn library for implementing decision tree regression.
- **sklearn.preprocessing.LabelEncoder**: A class from scikit-learn library for label encoding categorical variables.
- **sklearn.metrics.accuracy_score**: A function from scikit-learn library for computing accuracy scores.
- **sklearn.model_selection.train_test_split**: A function from scikit-learn library for splitting data into training and testing sets.
- **pickle**: A module for serializing and deserializing Python objects.


jupyter OverallPredictionModel Last Checkpoint: 15 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import pandas as pd
        import numpy as np
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.preprocessing import LabelEncoder
        from sklearn.metrics import accuracy_score
        from sklearn.model_selection import train_test_split
        import pickle
```

```
In [2]: df = pd.read_csv("Child_Growth.csv")
        df
```

Out[2]:

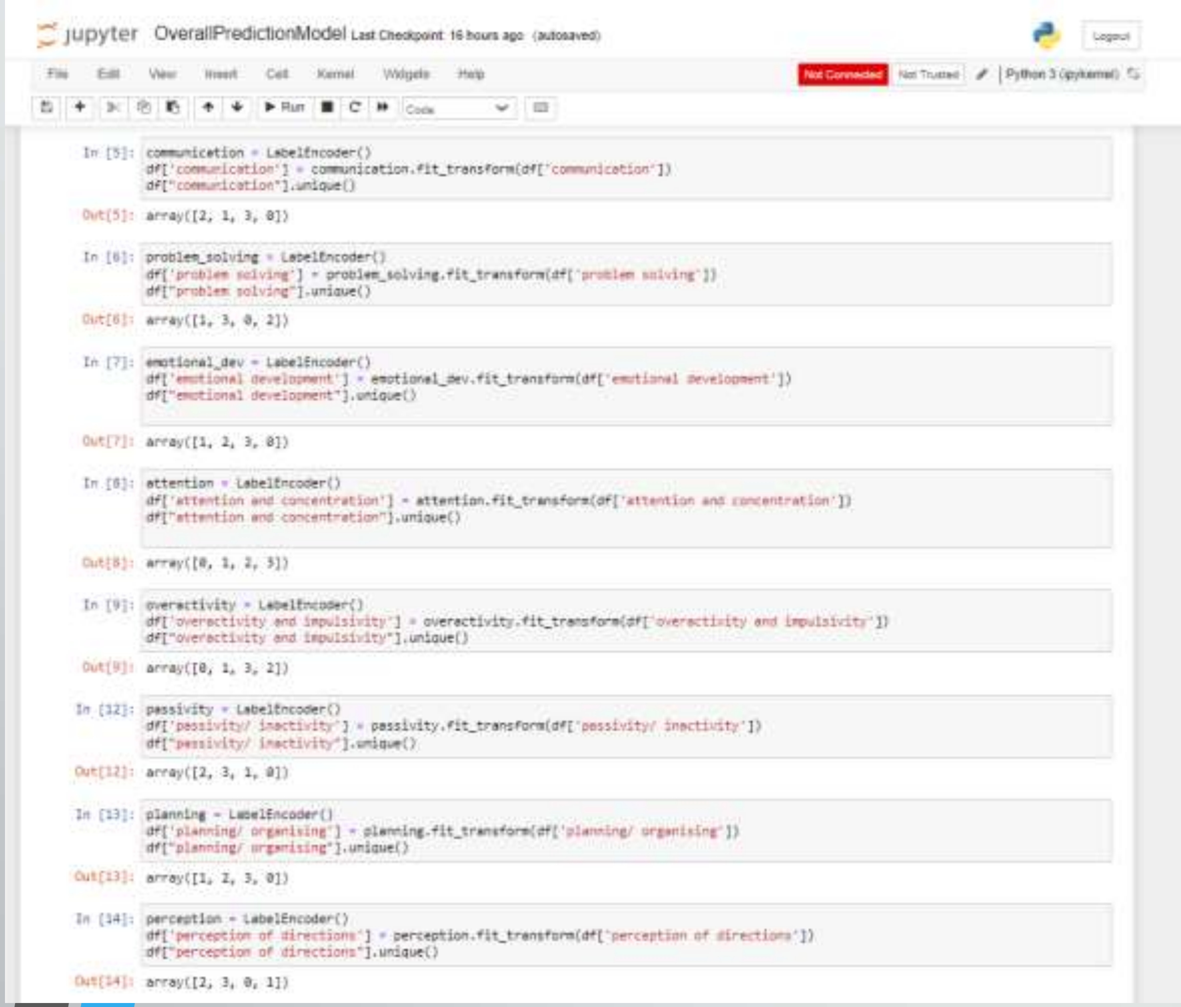
	weight	height	age	gross motor	fine motor	communication	problem solving	emotional development	attention and concentration	overactivity and impulsivity	passivity/ inactivity	planning/ organising	perception of directions	perceptior of visua forms anc figures
0	16	113	1	poor	excellent	good	fair	fair	excellent	excellent	good	fair	good	gooc
1	7	84	2	excellent	good	fair	poor	good	fair	fair	poor	good	poor	poo
2	17	106	1	poor	good	good	poor	poor	excellent	excellent	fair	poor	excellent	excellen
3	11	114	5	good	good	poor	excellent	poor	excellent	fair	fair	fair	good	poo
4	20	84	5	fair	excellent	poor	poor	excellent	fair	poor	fair	fair	poor	fai
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
79995	19	94	5	excellent	good	poor	excellent	excellent	fair	good	fair	poor	fair	excellen
79996	5	99	0	excellent	poor	excellent	poor	good	poor	poor	good	fair	excellent	fai
79997	4	115	5	fair	fair	poor	poor	good	fair	good	good	excellent	excellent	excellen
79998	10	83	3	fair	fair	poor	good	poor	excellent	poor	poor	fair	fair	gooc
79999	13	109	3	excellent	good	poor	good	good	good	poor	poor	excellent	excellent	excellen

80000 rows × 20 columns

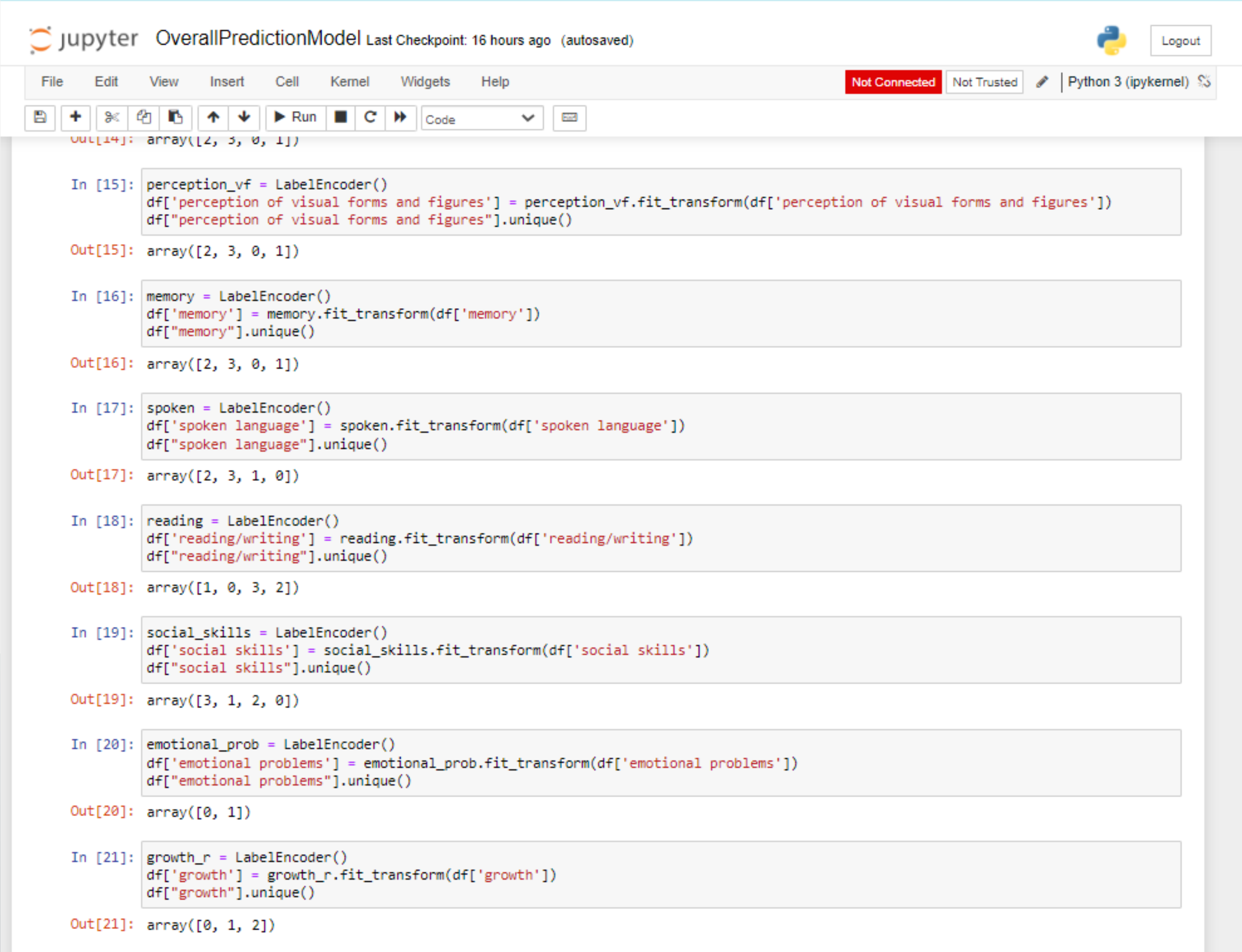
```
In [3]: gross_motor = LabelEncoder()
        df['gross motor'] = gross_motor.fit_transform(df['gross motor'])
        df["gross motor"].unique()

Out[3]: array([3, 0, 2, 1])
```

```
In [4]: fine_motor = LabelEncoder()
        df['fine motor'] = fine_motor.fit_transform(df['fine motor'])
        df["fine motor"].unique()

Out[4]: array([0, 2, 1, 3])
```

- The code loads a dataset from a CSV file using pd.read_csv from pandas library.
- The dataset is then filtered to select specific columns relevant to the model.
- Missing values are dropped using **dropna** function.
- Categorical variables are encoded using **LabelEncoder**.


jupyter OverallPredictionModel Last Checkpoint: 16 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Not Trusted
Python 3 (ipykernel)
Run
Code

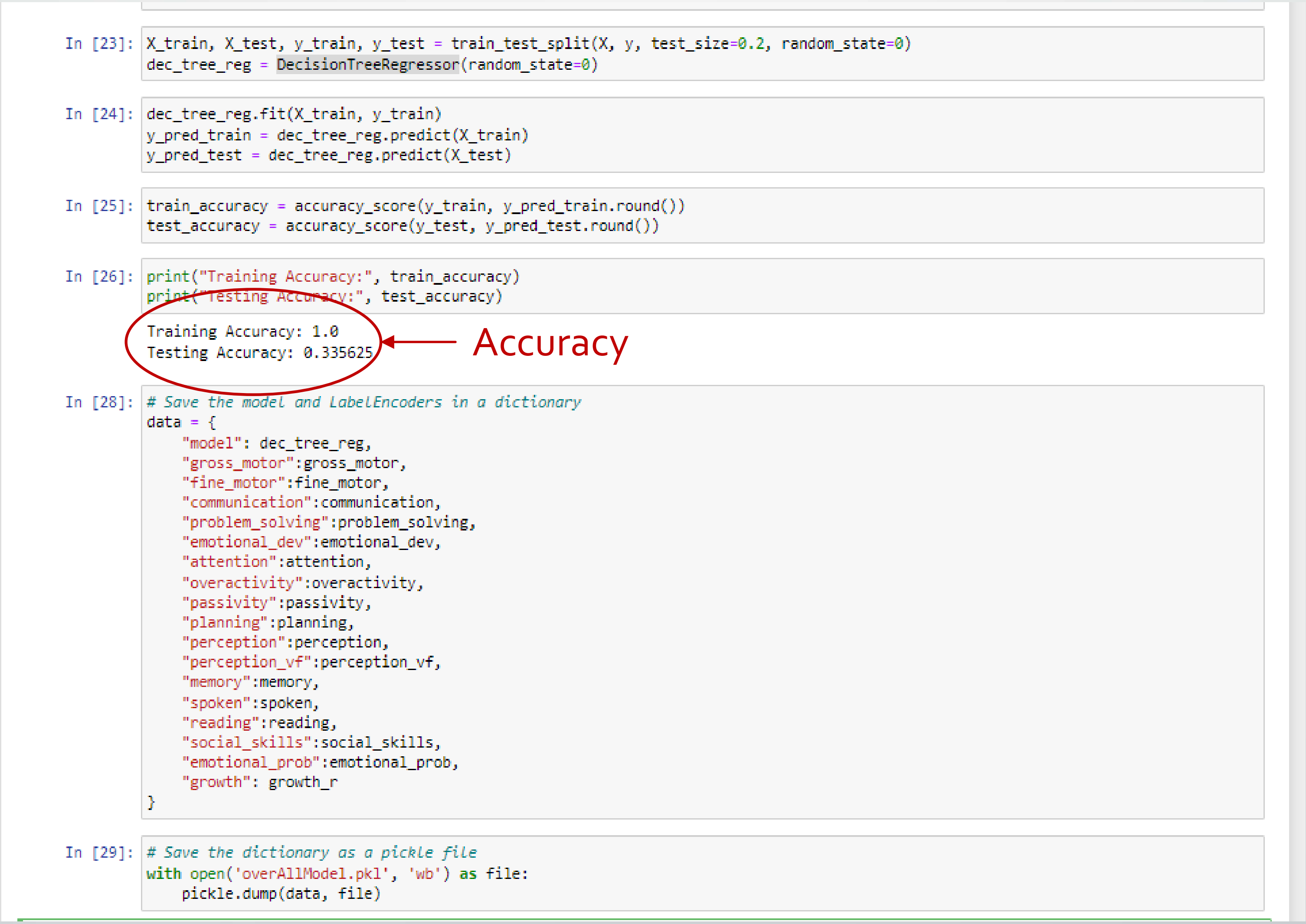
```
In [5]:  communication = LabelEncoder()
         df['communication'] = communication.fit_transform(df['communication'])
         df["communication"].unique()

Out[5]:  array([2, 1, 3, 0])

In [6]:  problem_solving = LabelEncoder()
         df['problem solving'] = problem_solving.fit_transform(df['problem solving'])
         df["problem solving"].unique()

Out[6]:  array([1, 3, 0, 2])

In [7]:  emotional_dev = LabelEncoder()
         df['emotional development'] = emotional_dev.fit_transform(df['emotional development'])
         df["emotional development"].unique()

Out[7]:  array([1, 2, 3, 0])

In [8]:  attention = LabelEncoder()
         df['attention and concentration'] = attention.fit_transform(df['attention and concentration'])
         df["attention and concentration"].unique()

Out[8]:  array([0, 1, 2, 3])

In [9]:  overactivity = LabelEncoder()
         df['overactivity and impulsivity'] = overactivity.fit_transform(df['overactivity and impulsivity'])
         df["overactivity and impulsivity"].unique()

Out[9]:  array([0, 1, 3, 2])

In [12]: passivity = LabelEncoder()
         df['passivity/ inactivity'] = passivity.fit_transform(df['passivity/ inactivity'])
         df["passivity/ inactivity"].unique()

Out[12]: array([2, 3, 1, 0])

In [13]: planning = LabelEncoder()
         df['planning/ organising'] = planning.fit_transform(df['planning/ organising'])
         df["planning/ organising"].unique()

Out[13]: array([1, 2, 3, 0])

In [14]: perception = LabelEncoder()
         df['perception of directions'] = perception.fit_transform(df['perception of directions'])
         df["perception of directions"].unique()

Out[14]: array([2, 3, 0, 1])
```


jupyter OverallPredictionModel Last Checkpoint: 16 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Not Trusted
Python 3 (ipykernel)
Run
Code

```
Out[14]: array([2, 3, 0, 1])

In [15]: perception_vf = LabelEncoder()
         df['perception of visual forms and figures'] = perception_vf.fit_transform(df['perception of visual forms and figures'])
         df["perception of visual forms and figures"].unique()

Out[15]: array([2, 3, 0, 1])

In [16]: memory = LabelEncoder()
         df['memory'] = memory.fit_transform(df['memory'])
         df["memory"].unique()

Out[16]: array([2, 3, 0, 1])

In [17]: spoken = LabelEncoder()
         df['spoken language'] = spoken.fit_transform(df['spoken language'])
         df["spoken language"].unique()

Out[17]: array([2, 3, 1, 0])

In [18]: reading = LabelEncoder()
         df['reading/writing'] = reading.fit_transform(df['reading/writing'])
         df["reading/writing"].unique()

Out[18]: array([1, 0, 3, 2])

In [19]: social_skills = LabelEncoder()
         df['social skills'] = social_skills.fit_transform(df['social skills'])
         df["social skills"].unique()

Out[19]: array([3, 1, 2, 0])

In [20]: emotional_prob = LabelEncoder()
         df['emotional problems'] = emotional_prob.fit_transform(df['emotional problems'])
         df["emotional problems"].unique()

Out[20]: array([0, 1])

In [21]: growth_r = LabelEncoder()
         df['growth'] = growth_r.fit_transform(df['growth'])
         df["growth"].unique()

Out[21]: array([0, 1, 2])
```

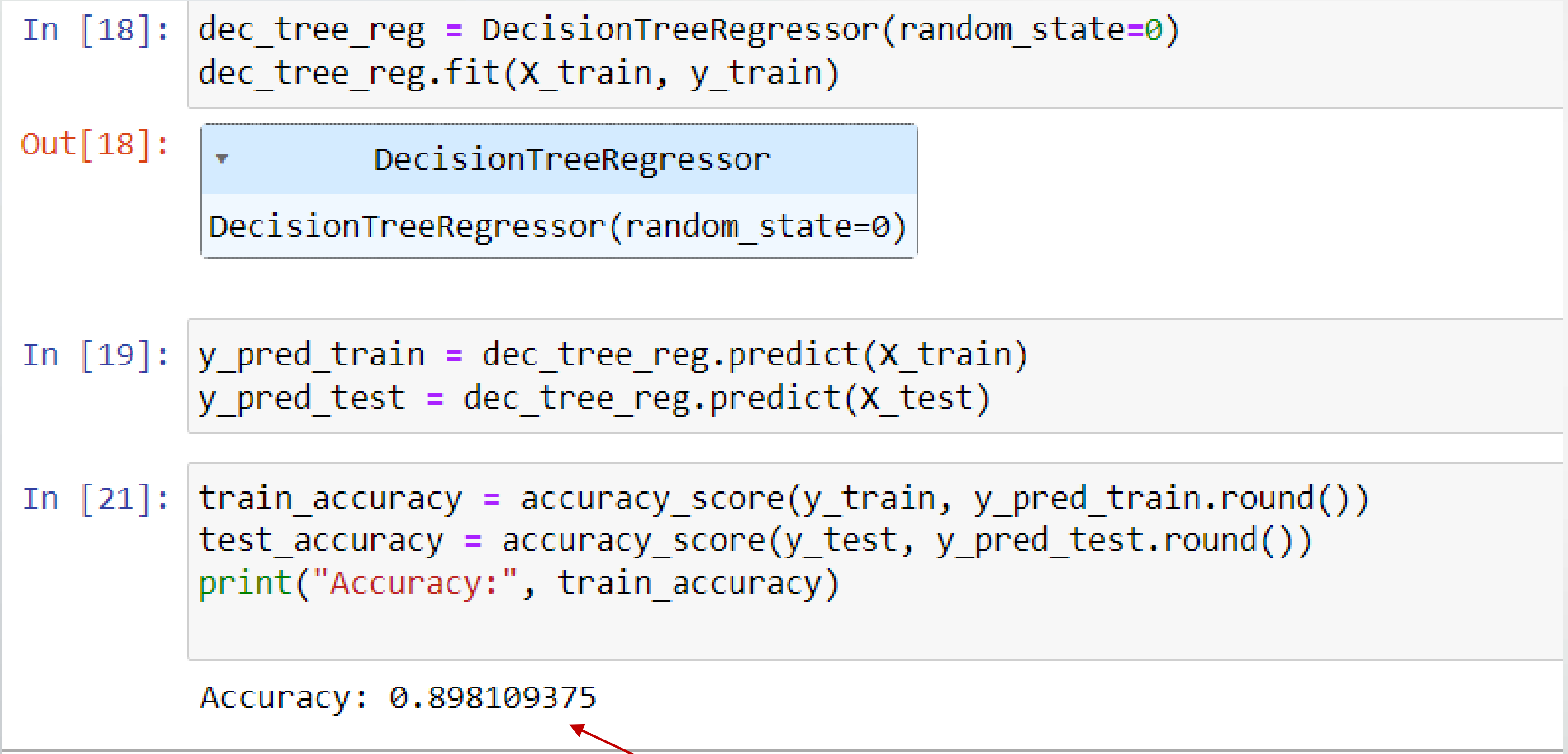
Data set reference - https://www.kaggle.com/datasets/salmanahmad1980/child-growth-measurements

- The code splits the preprocessed data into input features (X) and the target variable (y).
- The train_test_split function is used to divide the data into training and testing sets, with a test size of 20% and a random state of 0.
- **A decision tree regressor model** is instantiated using **DecisionTreeRegressor.**
- The model is trained using the training data with the fit method.



```
In [23]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
         dec_tree_reg = DecisionTreeRegressor(random_state=0)

In [24]: dec_tree_reg.fit(X_train, y_train)
         y_pred_train = dec_tree_reg.predict(X_train)
         y_pred_test = dec_tree_reg.predict(X_test)

In [25]: train_accuracy = accuracy_score(y_train, y_pred_train.round())
         test_accuracy = accuracy_score(y_test, y_pred_test.round())

In [26]: print("Training Accuracy:", train_accuracy)
         print("Testing Accuracy:", test_accuracy)

         Training Accuracy: 1.0
         Testing Accuracy: 0.335625

In [28]: # Save the model and LabelEncoders in a dictionary
         data = {
             "model": dec_tree_reg,
             "gross_motor":gross_motor,
             "fine_motor":fine_motor,
             "communication":communication,
             "problem_solving":problem_solving,
             "emotional_dev":emotional_dev,
             "attention":attention,
             "overactivity":overactivity,
             "passivity":passivity,
             "planning":planning,
             "perception":perception,
             "perception_vf":perception_vf,
             "memory":memory,
             "spoken":spoken,
             "reading":reading,
             "social_skills":social_skills,
             "emotional_prob":emotional_prob,
             "growth": growth_r
         }

In [29]: # Save the dictionary as a pickle file
         with open('overAllModel.pkl', 'wb') as file:
             pickle.dump(data, file)
```

Accuracy

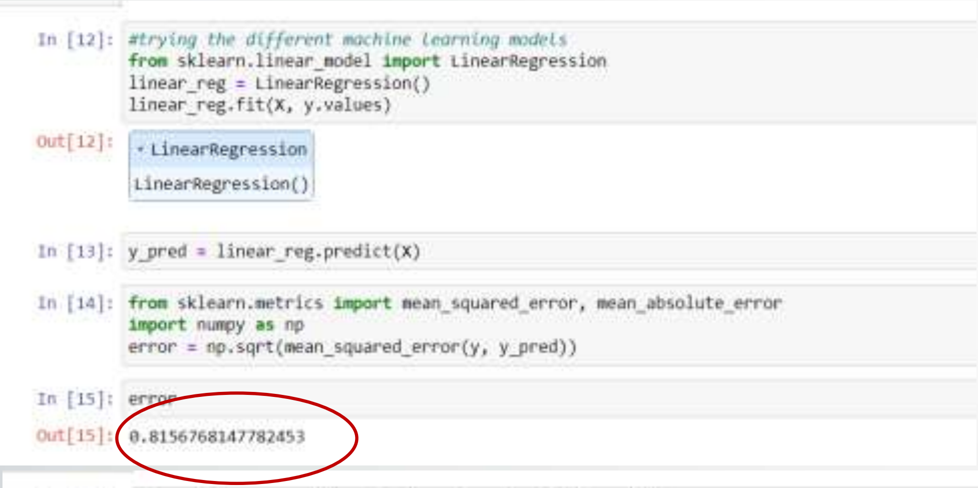


```
In [18]: dec_tree_reg = DecisionTreeRegressor(random_state=0)
         dec_tree_reg.fit(X_train, y_train)

Out[18]: ▾ DecisionTreeRegressor
         DecisionTreeRegressor(random_state=0)

In [19]: y_pred_train = dec_tree_reg.predict(X_train)
         y_pred_test = dec_tree_reg.predict(X_test)

In [21]: train_accuracy = accuracy_score(y_train, y_pred_train.round())
         test_accuracy = accuracy_score(y_test, y_pred_test.round())
         print("Accuracy:", train_accuracy)

         Accuracy: 0.898109375
```

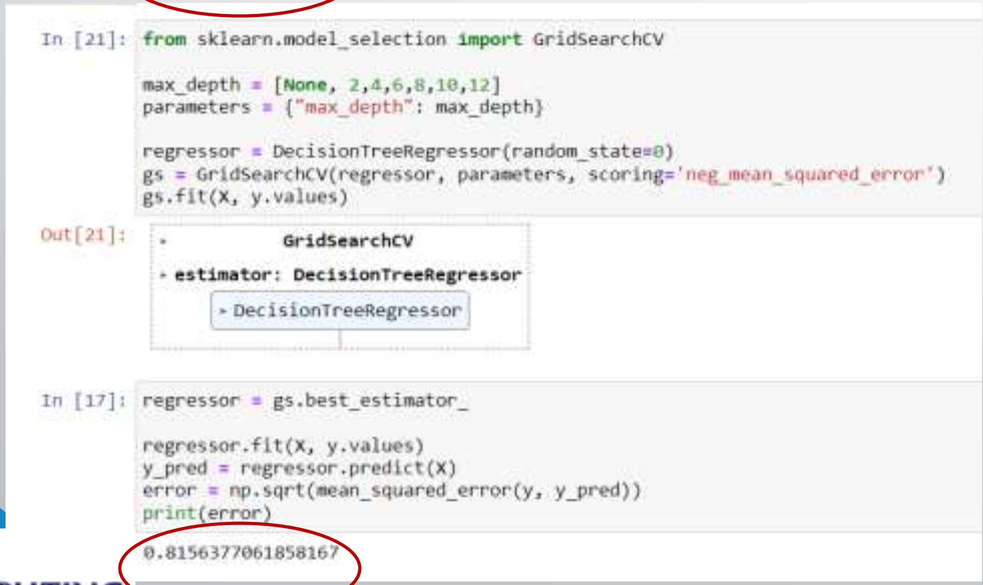
Model Accuracy – 90%

Selection of Decision Tree Regressor

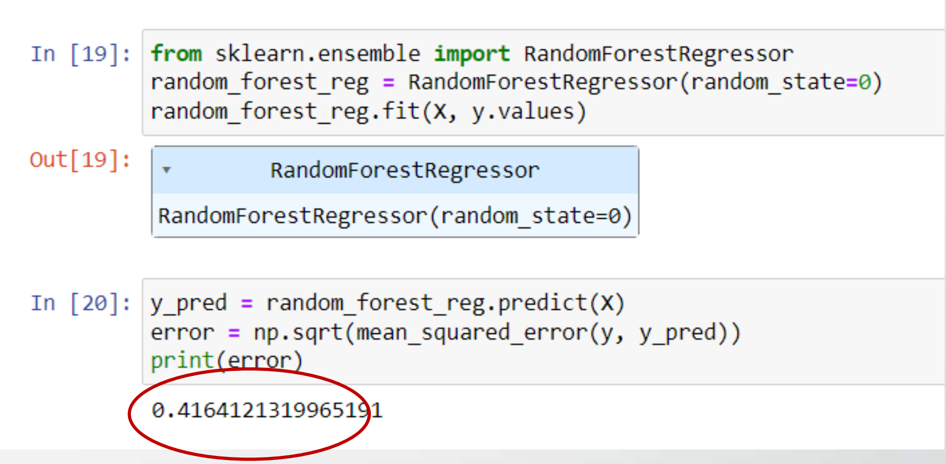
- Highest accuracy score and lowest error compared to other algorithms.



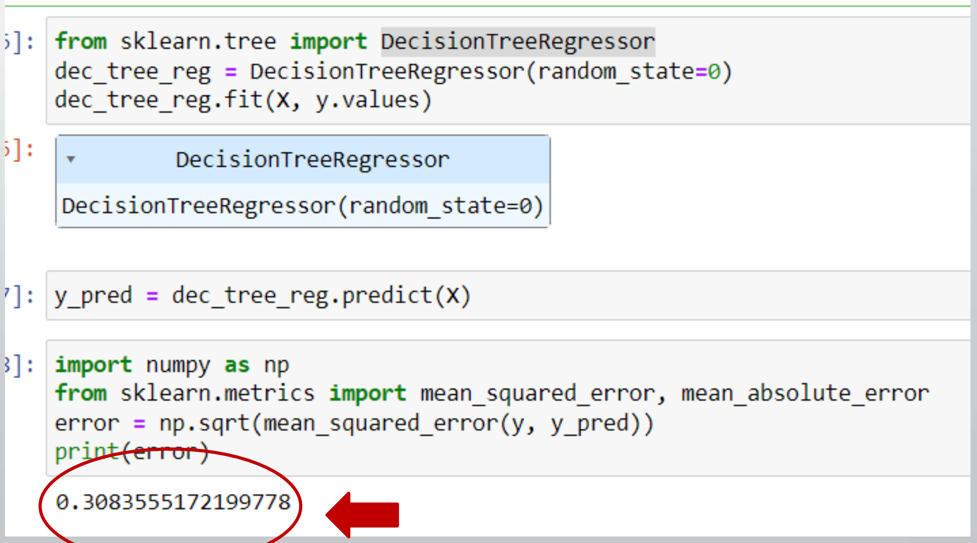
```
In [12]: #trying the different machine learning models
         from sklearn.linear_model import LinearRegression
         linear_reg = LinearRegression()
         linear_reg.fit(X, y.values)

Out[12]: ▾ LinearRegression
         LinearRegression()

In [13]: y_pred = linear_reg.predict(X)

In [14]: from sklearn.metrics import mean_squared_error, mean_absolute_error
         import numpy as np
         error = np.sqrt(mean_squared_error(y, y_pred))

In [15]: error
Out[15]: 0.8156768147782453
```



```
In [21]: from sklearn.model_selection import GridSearchCV

         max_depth = [None, 2,4,6,8,10,12]
         parameters = {"max_depth": max_depth}

         regressor = DecisionTreeRegressor(random_state=0)
         gs = GridSearchCV(regressor, parameters, scoring='neg_mean_squared_error')
         gs.fit(X, y.values)

Out[21]: ▸          GridSearchCV
         ▸ estimator: DecisionTreeRegressor
               ▸ DecisionTreeRegressor

In [17]: regressor = gs.best_estimator_

         regressor.fit(X, y.values)
         y_pred = regressor.predict(X)
         error = np.sqrt(mean_squared_error(y, y_pred))
         print(error)

         0.8156377061858167
```

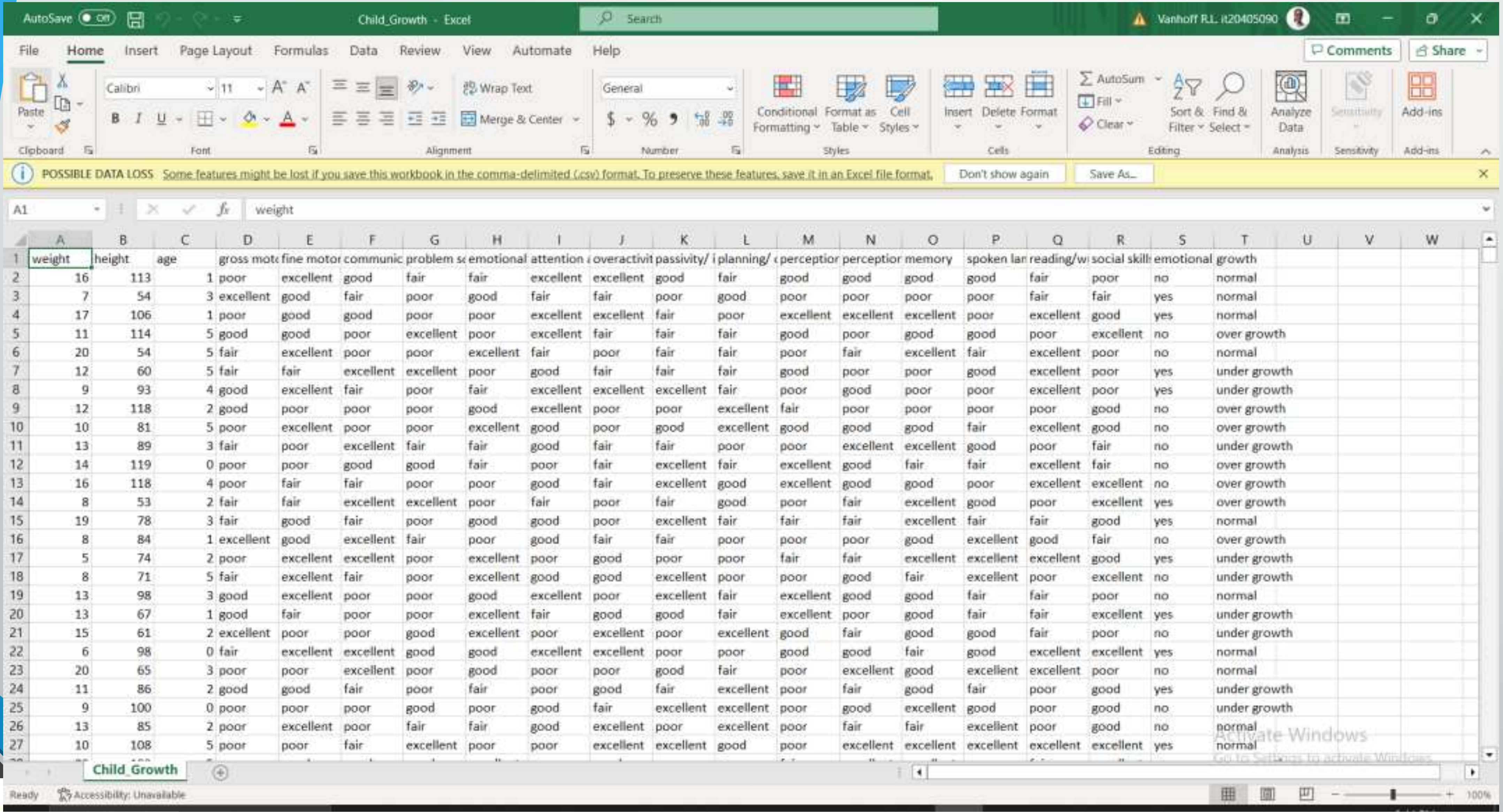


```
In [19]: from sklearn.ensemble import RandomForestRegressor
         random_forest_reg = RandomForestRegressor(random_state=0)
         random_forest_reg.fit(X, y.values)

Out[19]: ▾       RandomForestRegressor
         RandomForestRegressor(random_state=0)

In [20]: y_pred = random_forest_reg.predict(X)
         error = np.sqrt(mean_squared_error(y, y_pred))
         print(error)

         0.4164121319965191
```



```
5]: from sklearn.tree import DecisionTreeRegressor
    dec_tree_reg = DecisionTreeRegressor(random_state=0)
    dec_tree_reg.fit(X, y.values)

5]: ▾       DecisionTreeRegressor
    DecisionTreeRegressor(random_state=0)

7]: y_pred = dec_tree_reg.predict(X)

3]: import numpy as np
    from sklearn.metrics import mean_squared_error, mean_absolute_error
    error = np.sqrt(mean_squared_error(y, y_pred))
    print(error)

    0.3083555172199778
```

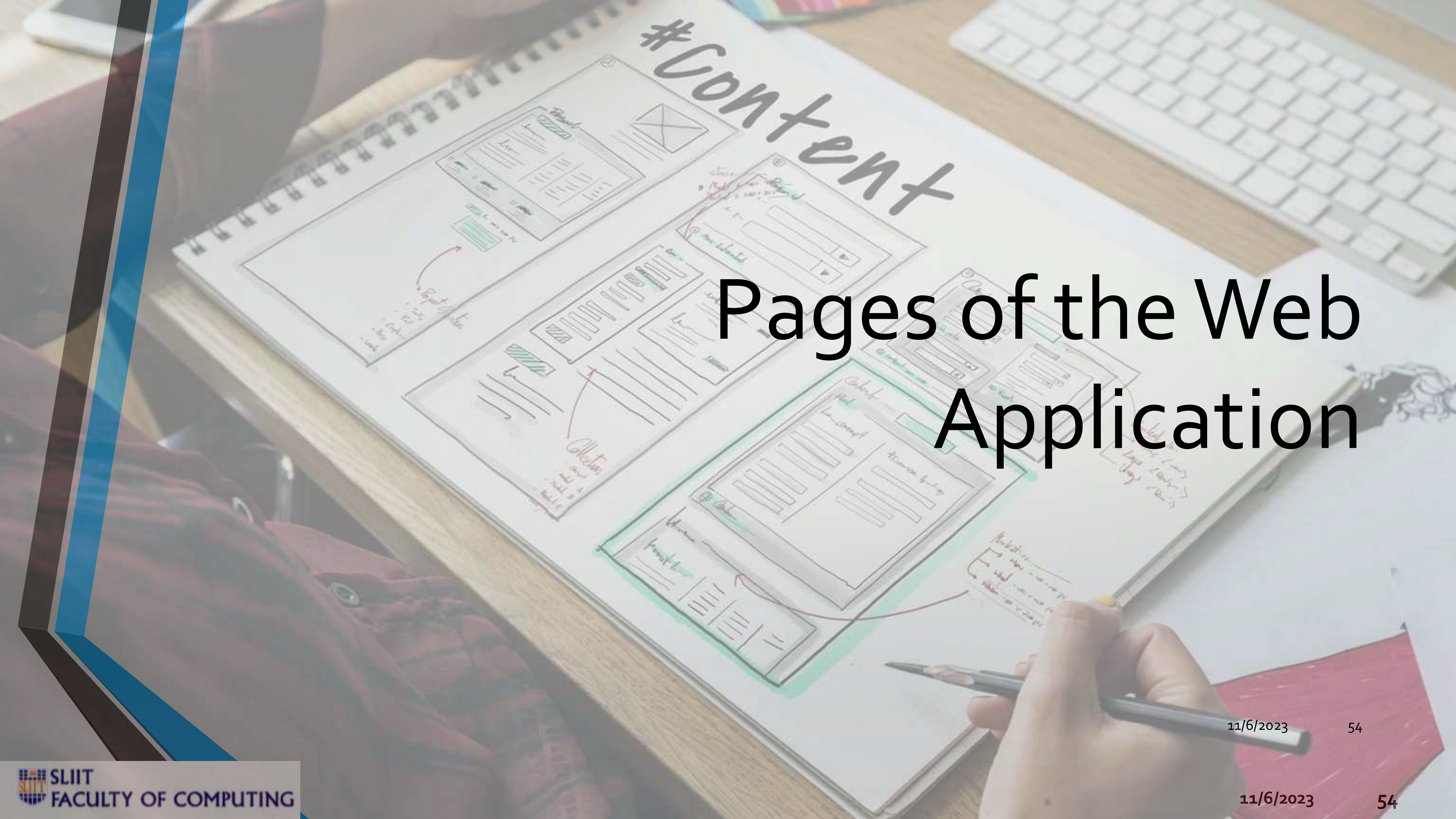
Dataset



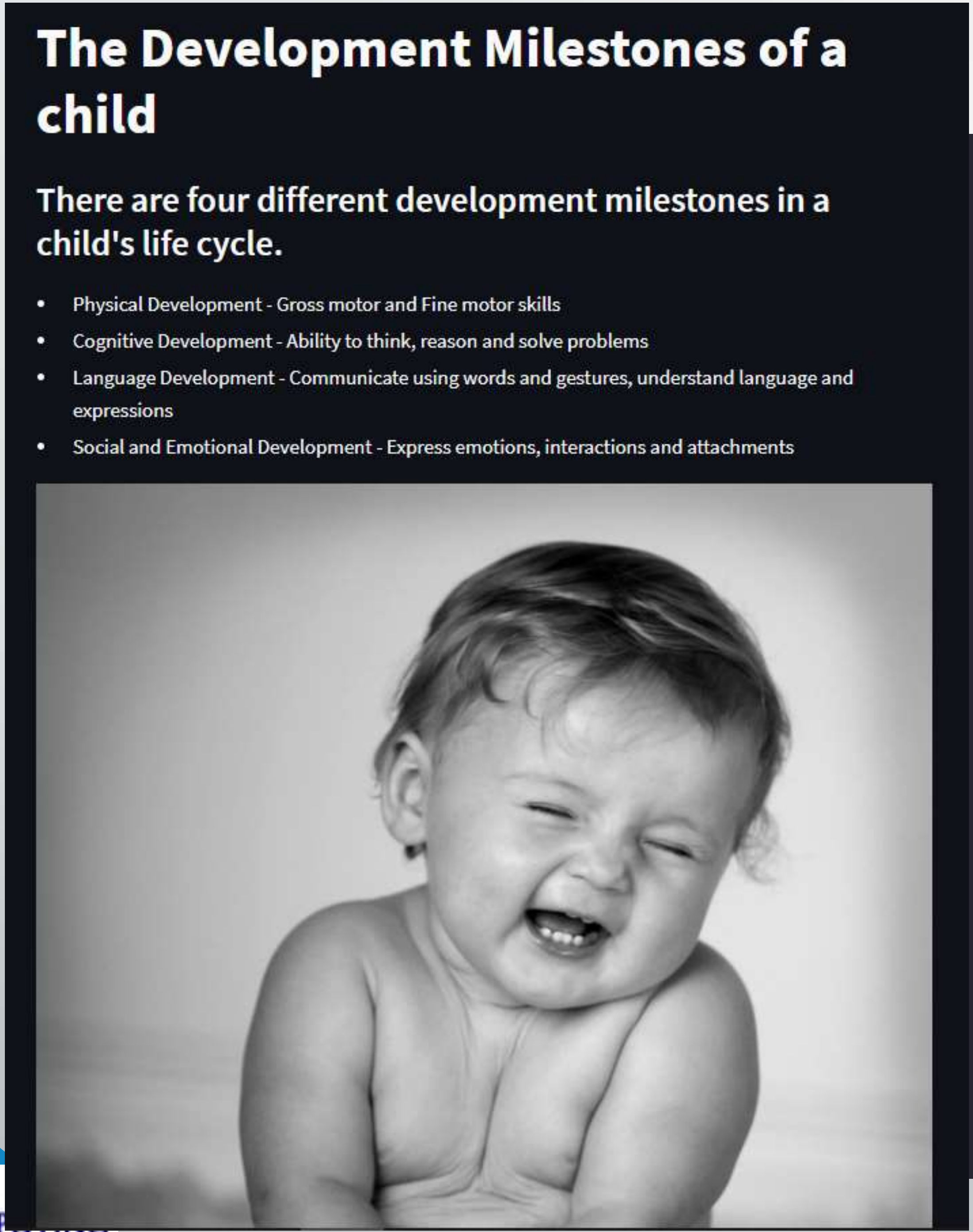
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 weight

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	weight	height	age	gross mot	fine motor	communic	problem s	emotional	attention	overactivit	passivity/ i	planning/	perceptior	perceptior	memory	spoken lan	reading/w	social skill	emotional	growth
2	16	113	1	poor	excellent	good	fair	fair	excellent	excellent	good	fair	good	good	good	good	fair	poor	no	normal
3	7	54	3	excellent	good	fair	poor	good	fair	fair	poor	good	poor	poor	poor	poor	fair	fair	yes	normal
4	17	106	1	poor	good	good	poor	poor	excellent	excellent	fair	poor	excellent	excellent	excellent	poor	excellent	good	yes	normal
5	11	114	5	good	good	poor	excellent	poor	excellent	fair	fair	fair	good	poor	good	good	poor	excellent	no	over growth
6	20	54	5	fair	excellent	poor	poor	excellent	fair	poor	fair	fair	poor	fair	excellent	fair	excellent	poor	no	normal
7	12	60	5	fair	fair	excellent	excellent	poor	good	fair	fair	fair	good	poor	poor	good	excellent	poor	yes	under growth
8	9	93	4	good	excellent	fair	poor	fair	excellent	excellent	excellent	fair	poor	good	poor	poor	excellent	poor	yes	under growth
9	12	118	2	good	poor	poor	poor	good	excellent	poor	poor	excellent	fair	poor	poor	poor	poor	good	no	over growth
10	10	81	5	poor	excellent	poor	poor	excellent	good	poor	good	excellent	good	good	good	fair	excellent	good	no	over growth
11	13	89	3	fair	poor	excellent	fair	fair	good	fair	fair	poor	poor	excellent	excellent	good	poor	fair	no	under growth
12	14	119	0	poor	poor	good	good	fair	poor	fair	excellent	fair	excellent	good	fair	fair	excellent	fair	no	over growth
13	16	118	4	poor	fair	fair	poor	poor	good	fair	excellent	good	excellent	good	good	poor	excellent	excellent	no	over growth
14	8	53	2	fair	fair	excellent	excellent	poor	fair	poor	fair	good	poor	fair	excellent	good	poor	excellent	yes	over growth
15	19	78	3	fair	good	fair	poor	good	good	poor	excellent	fair	fair	fair	excellent	fair	fair	good	yes	normal
16	8	84	1	excellent	good	excellent	fair	poor	good	fair	fair	poor	poor	poor	good	excellent	good	fair	no	over growth
17	5	74	2	poor	excellent	excellent	poor	excellent	poor	good	poor	poor	fair	fair	excellent	excellent	excellent	good	yes	under growth
18	8	71	5	fair	excellent	fair	poor	excellent	good	good	excellent	poor	poor	good	fair	excellent	poor	excellent	no	under growth
19	13	98	3	good	excellent	poor	poor	good	excellent	poor	excellent	fair	excellent	good	good	fair	fair	poor	no	normal
20	13	67	1	good	fair	poor	poor	excellent	fair	good	good	fair	excellent	poor	good	fair	fair	excellent	yes	under growth
21	15	61	2	excellent	poor	poor	good	excellent	poor	excellent	poor	excellent	good	fair	good	good	fair	poor	no	under growth
22	6	98	0	fair	excellent	excellent	good	good	excellent	excellent	poor	poor	good	good	fair	good	excellent	excellent	yes	normal
23	20	65	3	poor	poor	excellent	poor	good	poor	poor	good	fair	poor	excellent	good	excellent	excellent	poor	no	normal
24	11	86	2	good	good	fair	poor	fair	poor	good	fair	excellent	poor	fair	good	fair	poor	good	yes	under growth
25	9	100	0	poor	poor	poor	good	poor	good	fair	excellent	excellent	poor	good	excellent	good	poor	good	no	under growth
26	13	85	2	poor	excellent	poor	fair	fair	good	excellent	poor	excellent	poor	fair	fair	excellent	poor	good	no	normal
27	10	108	5	poor	poor	fair	excellent	poor	poor	excellent	excellent	good	poor	excellent	excellent	excellent	excellent	excellent	yes	normal



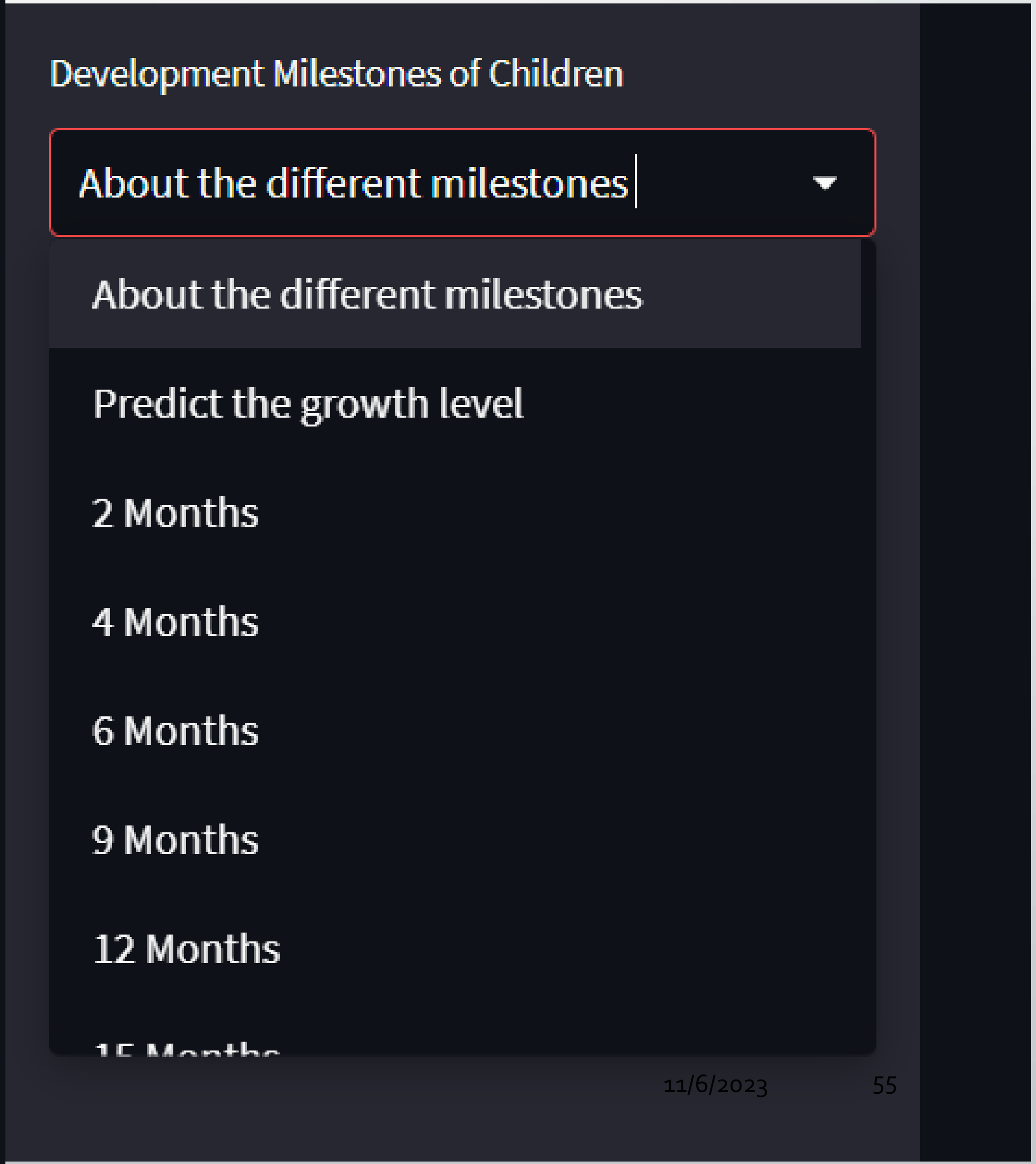
Pages of the Web Application



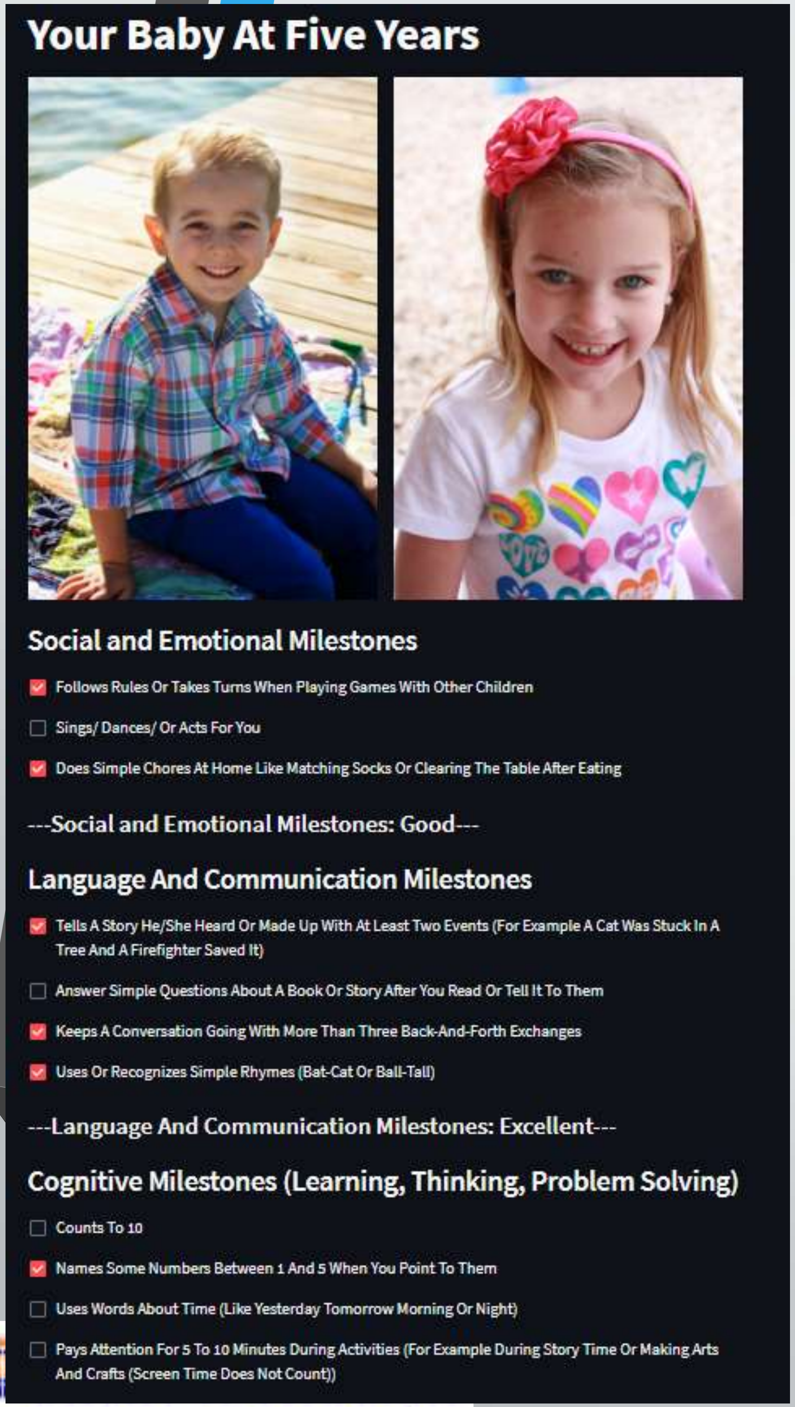
The Development Milestones of a child

There are four different development milestones in a child's life cycle.

- Physical Development - Gross motor and Fine motor skills
- Cognitive Development - Ability to think, reason and solve problems
- Language Development - Communicate using words and gestures, understand language and expressions
- Social and Emotional Development - Express emotions, interactions and attachments



Development Milestones of Children



Your Baby At Five Years

Social and Emotional Milestones

☑ Follows Rules Or Takes Turns When Playing Games With Other Children

☐ Sings/ Dances/ Or Acts For You

☑ Does Simple Chores At Home Like Matching Socks Or Clearing The Table After Eating

---Social and Emotional Milestones: Good---

Language And Communication Milestones

☑ Tells A Story He/She Heard Or Made Up With At Least Two Events (For Example A Cat Was Stuck In A Tree And A Firefighter Saved It)

☐ Answer Simple Questions About A Book Or Story After You Read Or Tell It To Them

☑ Keeps A Conversation Going With More Than Three Back-And-Forth Exchanges

☑ Uses Or Recognizes Simple Rhymes (Bat-Cat Or Ball-Tall)

---Language And Communication Milestones: Excellent---

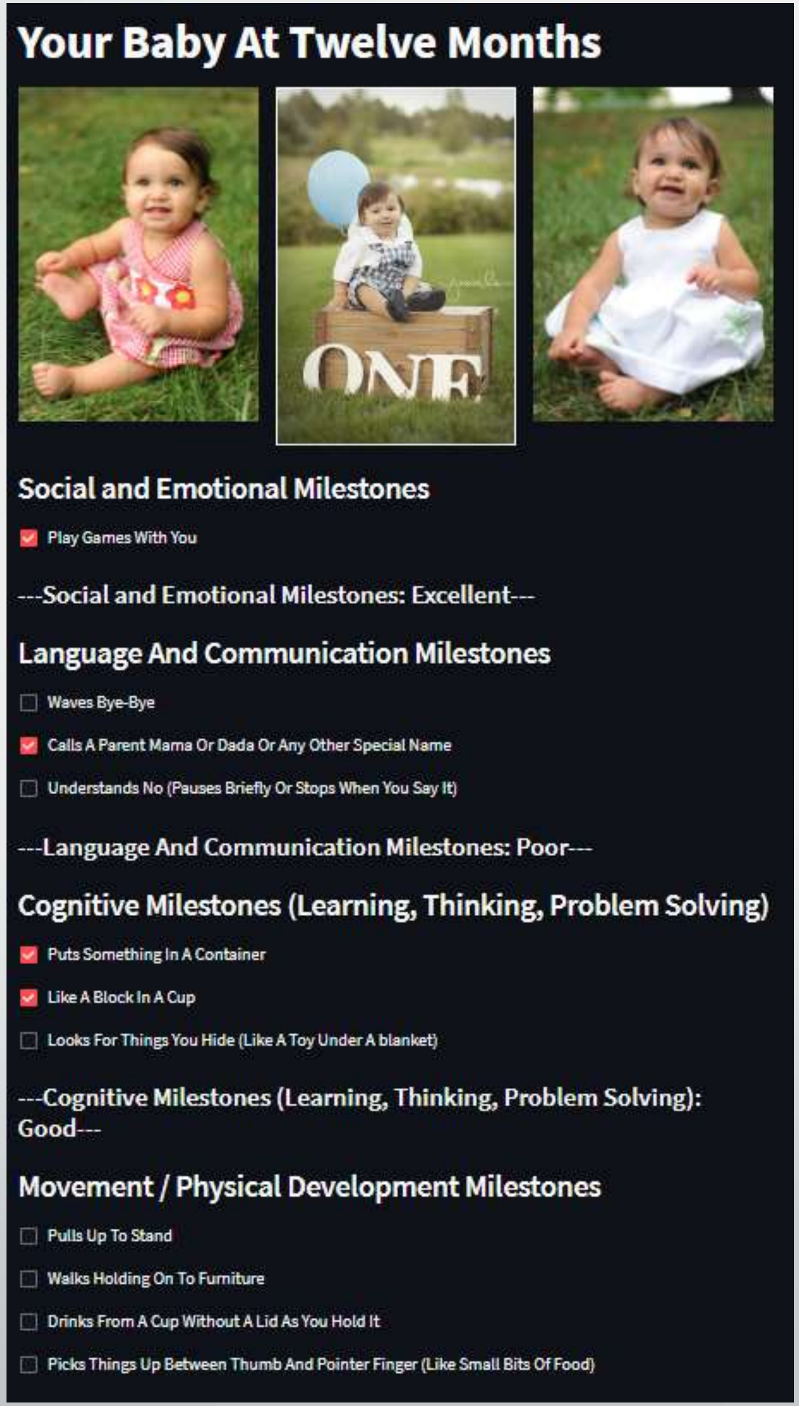
Cognitive Milestones (Learning, Thinking, Problem Solving)

☐ Counts To 10

☑ Names Some Numbers Between 1 And 5 When You Point To Them

☐ Uses Words About Time (Like Yesterday Tomorrow Morning Or Night)

☐ Pays Attention For 5 To 10 Minutes During Activities (For Example During Story Time Or Making Arts And Crafts (Screen Time Does Not Count))



Your Baby At Twelve Months

Social and Emotional Milestones

☑ Play Games With You

---Social and Emotional Milestones: Excellent---

Language And Communication Milestones

☐ Waves Bye-Bye

☑ Calls A Parent Mama Or Dada Or Any Other Special Name

☐ Understands No (Pauses Briefly Or Stops When You Say It)

---Language And Communication Milestones: Poor---

Cognitive Milestones (Learning, Thinking, Problem Solving)

☑ Puts Something In A Container

☑ Like A Block In A Cup

☐ Looks For Things You Hide (Like A Toy Under A blanket)

---Cognitive Milestones (Learning, Thinking, Problem Solving): Good---

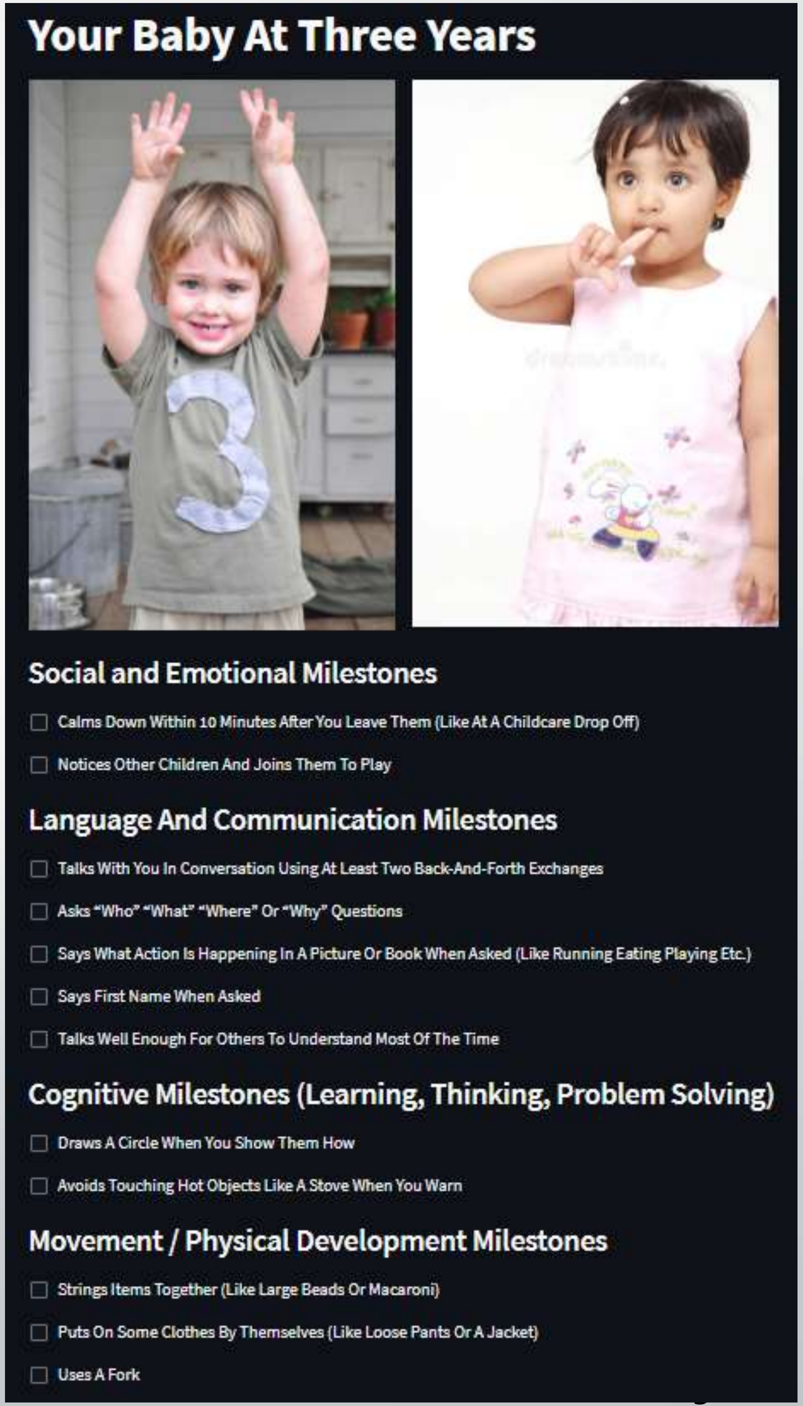
Movement / Physical Development Milestones

☐ Pulls Up To Stand

☐ Walks Holding On To Furniture

☐ Drinks From A Cup Without A Lid As You Hold It

☐ Picks Things Up Between Thumb And Pointer Finger (Like Small Bits Of Food)



Your Baby At Three Years

Social and Emotional Milestones

☐ Calms Down Within 10 Minutes After You Leave Them (Like At A Childcare Drop Off)

☐ Notices Other Children And Joins Them To Play

Language And Communication Milestones

☐ Talks With You In Conversation Using At Least Two Back-And-Forth Exchanges

☐ Asks "Who" "What" "Where" Or "Why" Questions

☐ Says What Action Is Happening In A Picture Or Book When Asked (Like Running Eating Playing Etc.)

☐ Says First Name When Asked

☐ Talks Well Enough For Others To Understand Most Of The Time

Cognitive Milestones (Learning, Thinking, Problem Solving)

☐ Draws A Circle When You Show Them How

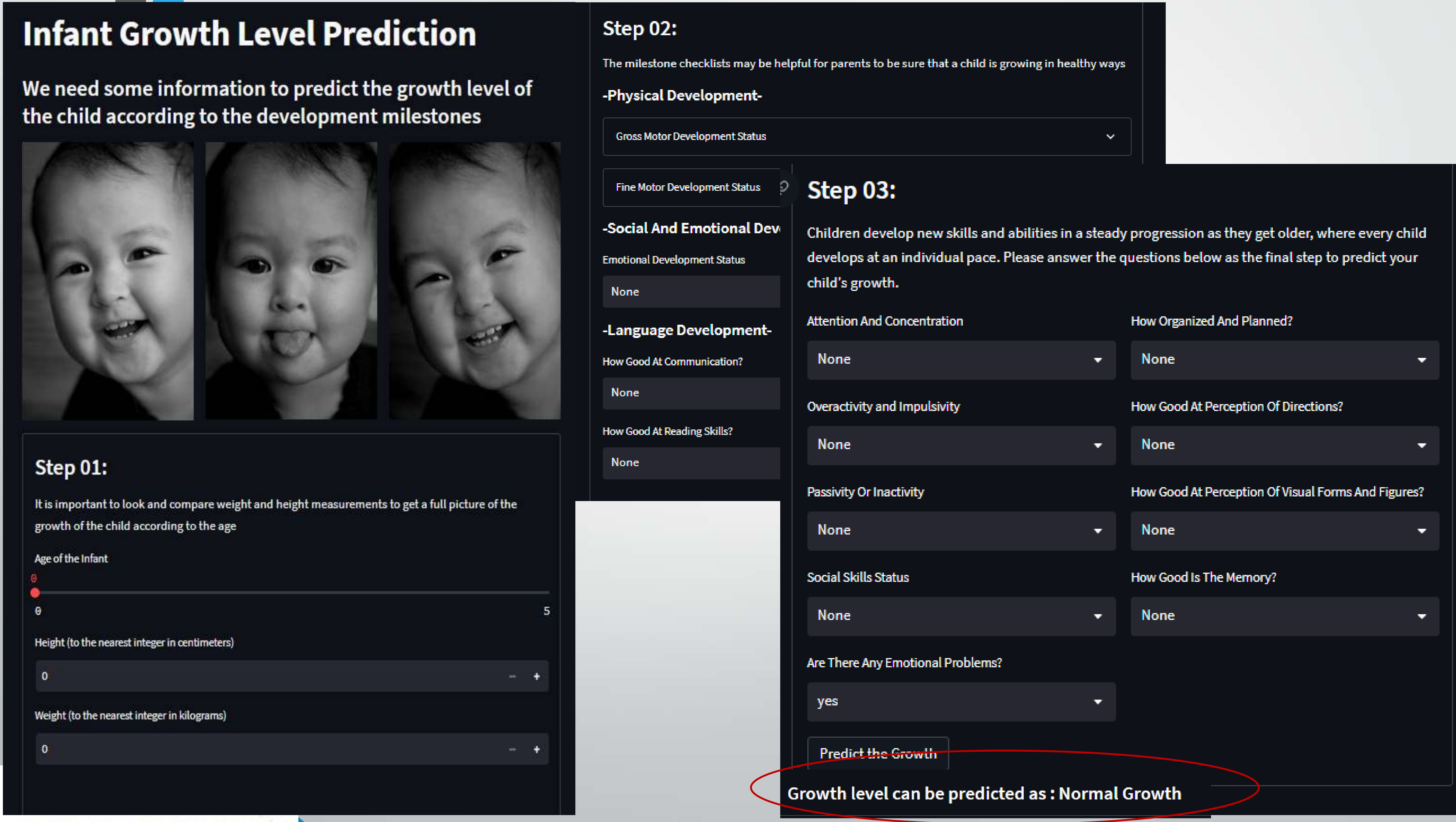
☐ Avoids Touching Hot Objects Like A Stove When You Warn

Movement / Physical Development Milestones

☐ Strings Items Together (Like Large Beads Or Macaroni)

☐ Puts On Some Clothes By Themselves (Like Loose Pants Or A Jacket)

☐ Uses A Fork



Infant Growth Level Prediction

We need some information to predict the growth level of the child according to the development milestones

Step 01:

It is important to look and compare weight and height measurements to get a full picture of the growth of the child according to the age

Age of the Infant: 0 (range 0 – 5)

Height (to the nearest integer in centimeters): 0

Weight (to the nearest integer in kilograms): 0

Step 02:

The milestone checklists may be helpful for parents to be sure that a child is growing in healthy ways

-Physical Development-

Gross Motor Development Status

Fine Motor Development Status

-Social And Emotional Dev

Emotional Development Status: None

-Language Development-

How Good At Communication?: None

How Good At Reading Skills?: None

Step 03:

Children develop new skills and abilities in a steady progression as they get older, where every child develops at an individual pace. Please answer the questions below as the final step to predict your child's growth.

Attention And Concentration: None

How Organized And Planned?: None

Overactivity and Impulsivity: None

How Good At Perception Of Directions?: None

Passivity Or Inactivity: None

How Good At Perception Of Visual Forms And Figures?: None

Social Skills Status: None

How Good Is The Memory?: None

Are There Any Emotional Problems?: yes

Growth level can be predicted as : Normal Growth

IT20156374| Fernando A.P

Infant Sickness Identification through Video Processing

Software Engineering

Background and Literature

Introduction and Background

- Infant developmental abnormalities need to be detected early in order to be treated quickly and achieve better results.
- In low-resource situations, such as Sri Lanka, access to early intervention services may be restricted.
- The focus of this research is to develop a system that utilizes video processing techniques to identify abnormal behaviors in infants.
- The system enables parents to record videos of their infants and provides guidance on capturing specific behaviors for analysis.
- By employing image processing and video processing techniques, the system analyzes the recorded videos to identify behaviors of infants.

Background and Literature cont.

- Early research indicates that newborns with ASD might demonstrate significant behavioral changes throughout the early stages of development, particularly if they spend time in the NICU.but they are more concerned about the 10 months. [1]
- This study focused on babies who were born very early and had very low birth weights. It was shown that these premature infants had large gray matter and cerebrospinal fluid volume decreases in their brains, which were connected to developmental problems in the first year of life.[2]
- This study concentrated on full-term infants at 2 years of corrected age and very preterm infants (born before 30 weeks of gestation or with a birth weight under 1,250 grams). It sought to contrast their social and emotional challenges and prowess. This study emphasizes the possibility of behavior issues and early difficulties with social and emotional development in very preterm newborns.[3]
- This study analyzes a novel technique termed the assessment of general movements (GMs) as a means to predict developmental abnormalities in infants, such as cerebral palsy (CP), and addresses the significance of early diagnosis of these diseases. The study shows that CP can be strongly predicted by the quality of GMs in newborns at two to four months post-term. [4]

Objectives

Main Objective - :

To detection and development of a system to support parents by recognizing and informing them of problematic behaviors in babies using video processing techniques.

Sub Objective - :

To enhanced monitoring: The system allows parents to actively monitor their infants for potential illnesses.

To early detection: By capturing and analyzing abnormal behaviors, this can help to early detection of sicknesses.

To peace of mind: This system empowers parents with a valuable tool to ensure the well-being of their infants.

Research Problem

What are the abnormal behaviors can predict?

e.g **Autism Spectrum Disorder (ASD) in Infants**

1. Social Interaction

Normally, Infants interact socially with their parents by making eye contact, smiling responsively, and responding to their names(Normal behavior). But Abnormally, Infants with ASD may exhibit limited engagement with others, avoid eye contact, and show less interest in social relationships.

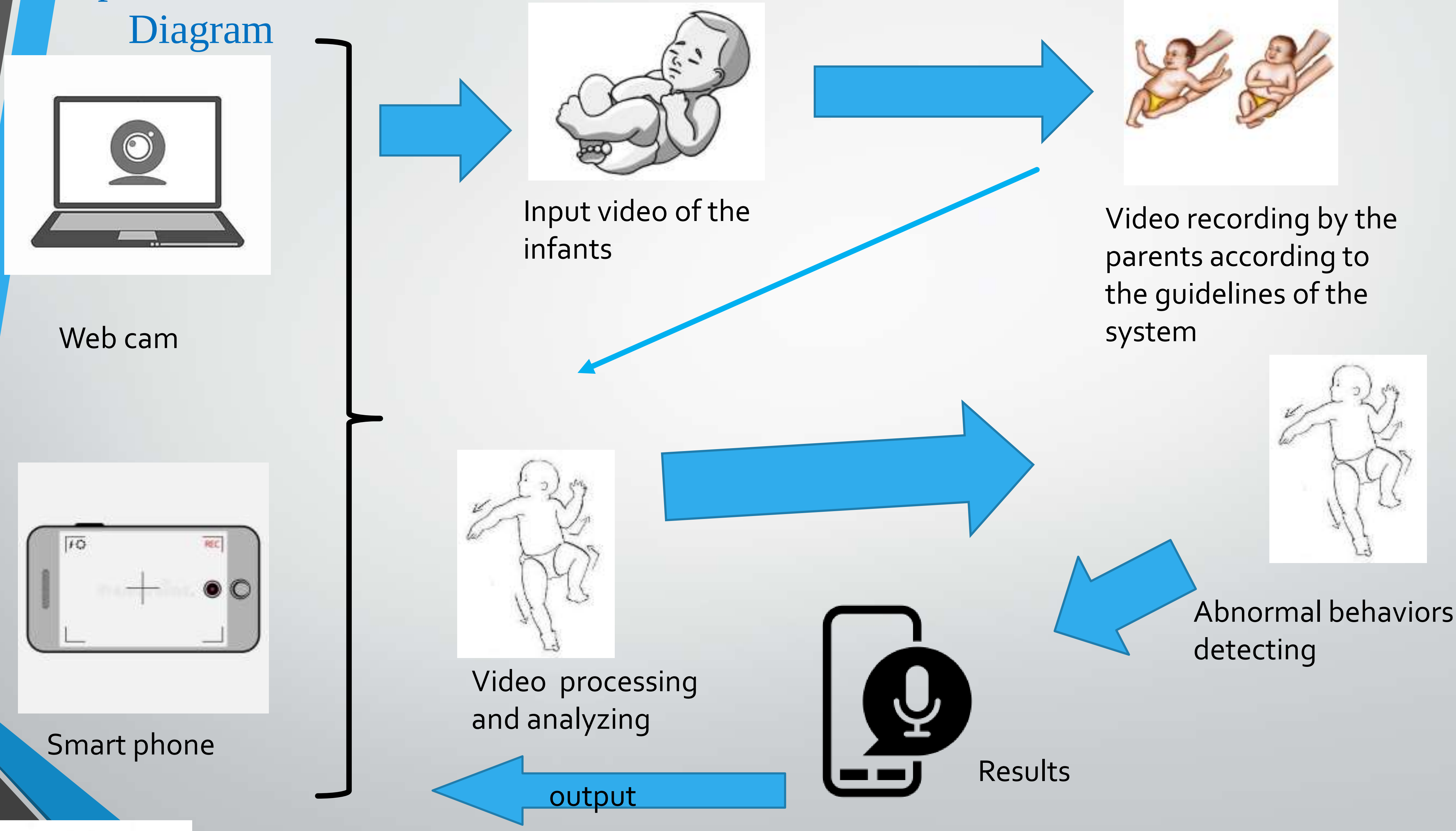
2. Communication:

Normal Behavior: Babies often start babbling, copying sounds, and trying to communicate their needs through gestures and vocalizations.

Abnormal Behavior: might exhibit delays in speech development, limited babbling, and reduced attempts to communicate or share attention

Likewise can be identify via recorded videos, the behaviors of communication, Social Responsiveness, Eye Contact.

Component Overview Diagram



Web cam

Smart phone

Input video of the infants

Video recording by the parents according to the guidelines of the system

Video processing and analyzing

Abnormal behaviors detecting

Results

output

Methodology

Datasets – Video datasets with abnormal and normal behaviors of babies

Prediction Model

- ❑ Technologies Involve – **Python**, **OpenCV (cv2), NumPy, TensorFlow, Keras, Matplotlib, MoviePy, scikit-learn (sklearn), pafy and Various Python libraries for file and data manipulation**
- ❑
- ❑ **TensorFlow/Keras, which enables the development of deep learning models, and OpenCV, which processes video frames, form the foundation of this approach. The LRCN (Long-term Recurrent Convolutional Networks) model architecture is employed in the code, which seems to be a component of a video-based classification challenge. In order to classify sequences of video frames, this model mixes recurrent neural networks (RNNs) and convolutional neural networks (CNNs).**

Methodology cont.

System Implementation

- ❑ Video recording and processing with Database Connection
- ❑ Make Prediction

Video recording and processing with Database Connection

❑

- ❑ Technologies Involve
- ▪ **OpenCV (cv2),**
- ▪ **Streamlit (basically for frontend development)**
- ▪ **Python Standard Libraries eg: subprosess, datetime**
- ▪ **Video Encoding Libraries (e.g., H.264)**

- ❑ **MongoDB (MongoDB Atlas) for Database Communication**

❑ Make Prediction

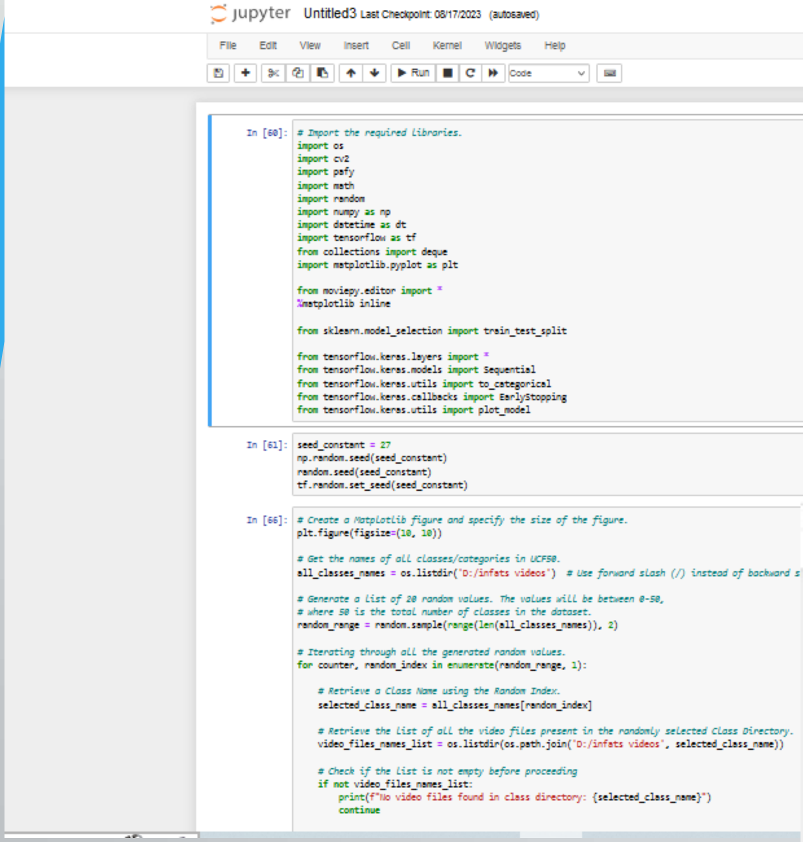
❑ Technologies Involve

Python, OpenCV (cv2), NumPy, TensorFlow, Keras, PIL (Python Imaging Library), HTML/CSS (via Streamlit)

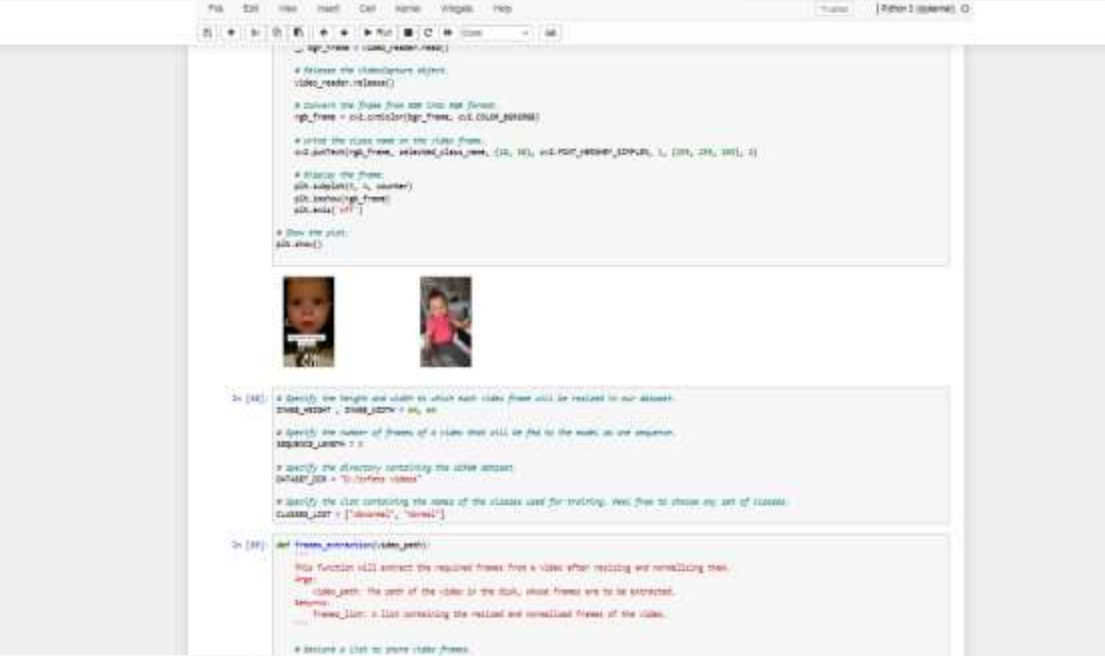
Jupyter Notebook
Visual Studio Code

Developed solution

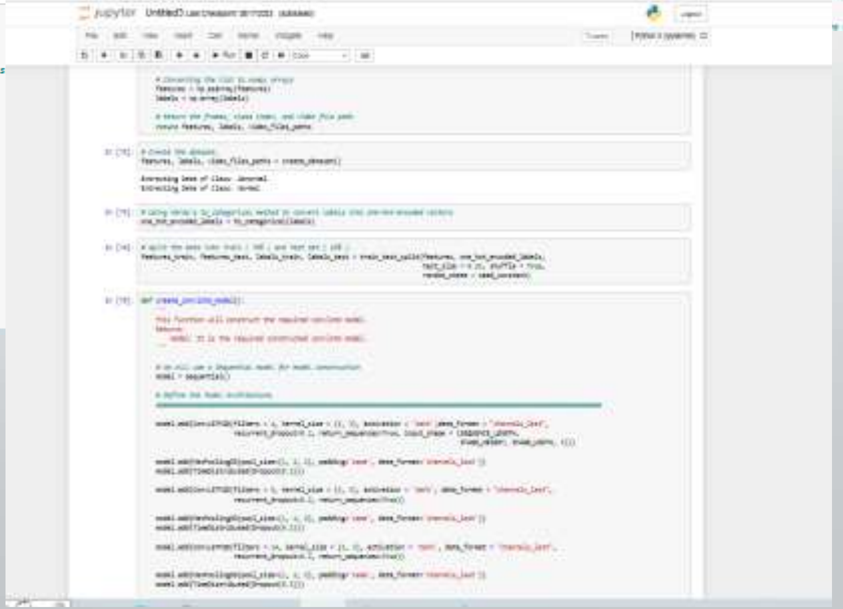
Dataset Training and model for make prediction


jupyter Untitled3 Last Checkpoint: 08/17/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel) ○
Run
Code

```
In [60]: # Import the required libraries.
         import os
         import cv2
         import pafy
         import math
         import random
         import numpy as np
         import datetime as dt
         import tensorflow as tf
         from collections import deque
         import matplotlib.pyplot as plt

         from moviepy.editor import *
         %matplotlib inline

         from sklearn.model_selection import train_test_split

         from tensorflow.keras.layers import *
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.utils import to_categorical
         from tensorflow.keras.callbacks import EarlyStopping
         from tensorflow.keras.utils import plot_model

In [61]: seed_constant = 27
         np.random.seed(seed_constant)
         random.seed(seed_constant)
         tf.random.set_seed(seed_constant)

In [66]: # Create a Matplotlib figure and specify the size of the figure.
         plt.figure(figsize=(10, 10))

         # Get the names of all classes/categories in UCF50.
         all_classes_names = os.listdir('D:/infats videos')  # Use forward slash (/) instead of backward s

         # Generate a list of 20 random values. The values will be between 0-50,
         # where 50 is the total number of classes in the dataset.
         random_range = random.sample(range(len(all_classes_names)), 2)

         # Iterating through all the generated random values.
         for counter, random_index in enumerate(random_range, 1):

             # Retrieve a Class Name using the Random Index.
             selected_class_name = all_classes_names[random_index]

             # Retrieve the list of all the video files present in the randomly selected Class Directory.
             video_files_names_list = os.listdir(os.path.join('D:/infats videos', selected_class_name))

             # Check if the list is not empty before proceeding
             if not video_files_names_list:
                 print(f"No video files found in class directory: {selected_class_name}")
                 continue
```



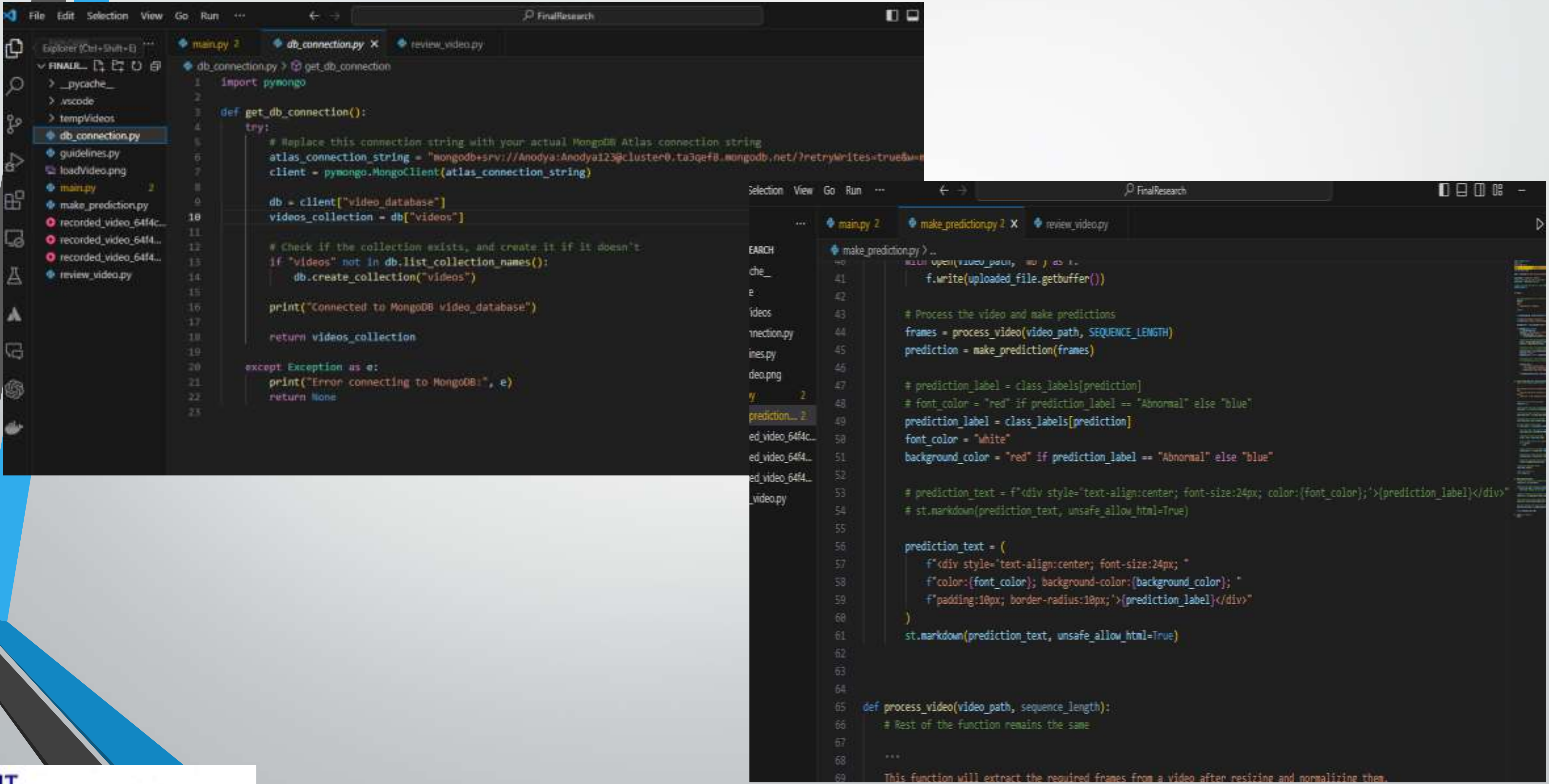
```
    _, bgr_frame = video_reader.read()

    # Release the VideoCapture object.
    video_reader.release()

    # Convert the frame from BGR into RGB format.
    rgb_frame = cv2.cvtColor(bgr_frame, cv2.COLOR_BGR2RGB)

    # Write the class name on the video frame.
    cv2.putText(rgb_frame, selected_class_name, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)

    # Display the frame.
    plt.subplot(5, 4, counter)
    plt.imshow(rgb_frame)
    plt.axis('off')

# Show the plot.
plt.show()

In [[illegible]]: # Specify the height and width to which each video frame will be resized in our dataset.
         IMAGE_HEIGHT , IMAGE_WIDTH = 64, 64

         # Specify the number of frames of a video that will be fed to the model as one sequence.
         SEQUENCE_LENGTH = 5

         # Specify the directory containing the UCF50 dataset.
         DATASET_DIR = "D:/infats videos"

         # Specify the list containing the names of the classes used for training. Feel free to choose any set of classes.
         CLASSES_LIST = ["abnormal", "normal"]

In [[illegible]]: def frames_extraction(video_path):
             '''
             This function will extract the required frames from a video after resizing and normalizing them.
             Args:
                 video_path: The path of the video in the disk, whose frames are to be extracted.
             Returns:
                 frames_list: A list containing the resized and normalized frames of the video.
             '''

             # Declare a list to store video frames.
```


jupyter Untitled3 Last Checkpoint: 08/17/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel) ○
Run
Code

```
    # Converting the list to numpy arrays
    features = np.asarray(features)
    labels = np.array(labels)

    # Return the frames, class index, and video file path.
    return features, labels, video_files_paths

In [72]: # Create the dataset.
         features, labels, video_files_paths = create_dataset()

         Extracting Data of Class: Abnormal
         Extracting Data of Class: Normal

In [73]: # Using Keras's to_categorical method to convert labels into one-hot-encoded vectors
         one_hot_encoded_labels = to_categorical(labels)

In [74]: # Split the Data into Train ( 75% ) and Test Set ( 25% ).
         features_train, features_test, labels_train, labels_test = train_test_split(features, one_hot_encoded_labels,
                                                                                     test_size = 0.25, shuffle = True,
                                                                                     random_state = seed_constant)

In [75]: def create_convlstm_model():
             '''
             This function will construct the required convlstm model.
             Returns:
                 model: It is the required constructed convlstm model.
             '''

             # We will use a Sequential model for model construction
             model = Sequential()

             # Define the Model Architecture.
             ########################################################################################################

             model.add(ConvLSTM2D(filters = 4, kernel_size = (3, 3), activation = 'tanh',data_format = "channels_last",
                                  recurrent_dropout=0.2, return_sequences=True, input_shape = (SEQUENCE_LENGTH,
                                                                                               IMAGE_HEIGHT, IMAGE_WIDTH, 3)))

             model.add(MaxPooling3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))
             model.add(TimeDistributed(Dropout(0.2)))

             model.add(ConvLSTM2D(filters = 8, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",
                                  recurrent_dropout=0.2, return_sequences=True))

             model.add(MaxPooling3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))
             model.add(TimeDistributed(Dropout(0.2)))

             model.add(ConvLSTM2D(filters = 14, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",
                                  recurrent_dropout=0.2, return_sequences=True))

             model.add(MaxPooling3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))
             model.add(TimeDistributed(Dropout(0.2)))
```

Developed solution cont.

- **Main application**

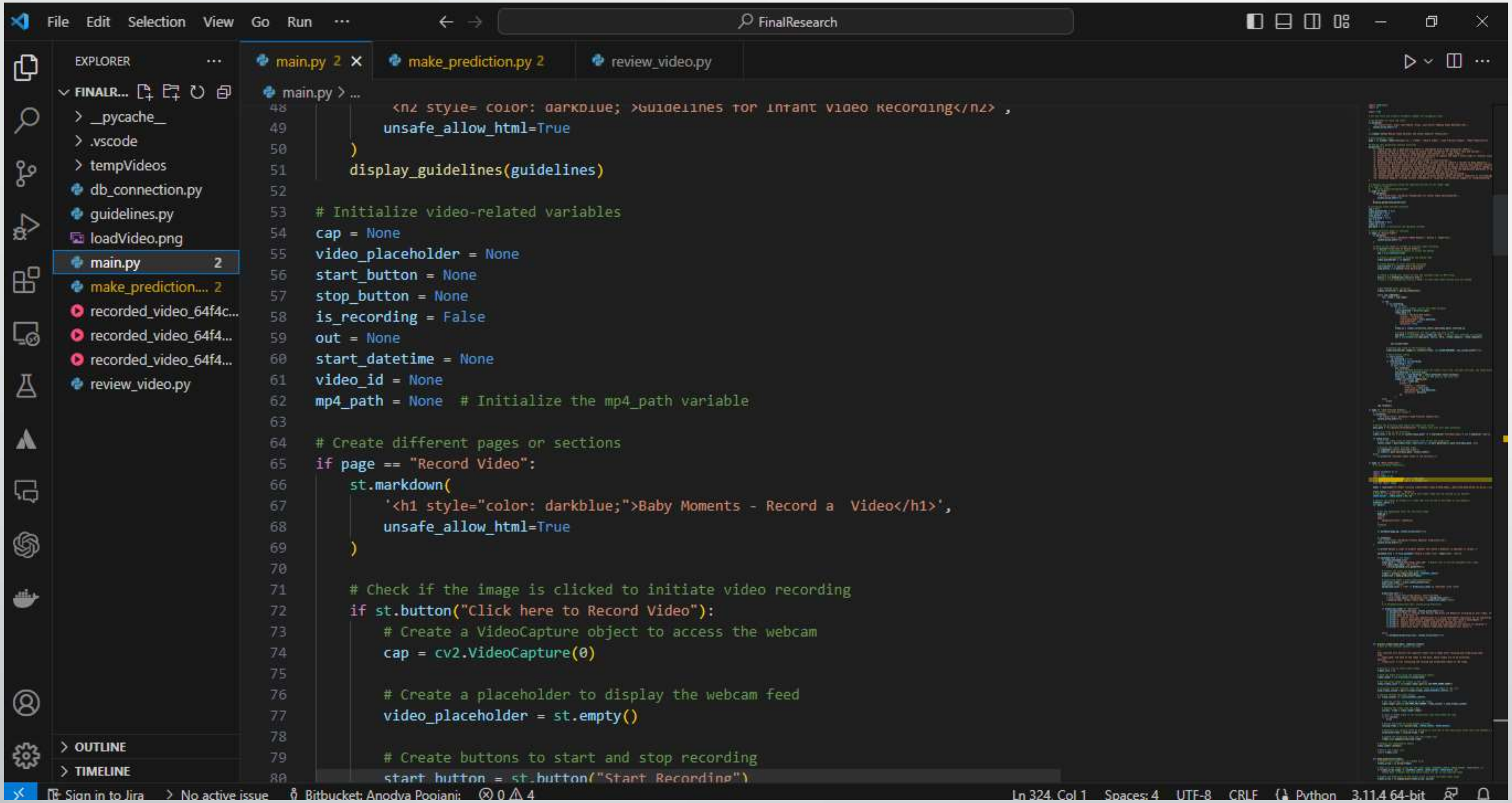


```python
import pymongo

def get_db_connection():
    try:
        # Replace this connection string with your actual MongoDB Atlas connection string
        atlas_connection_string = "mongodb+srv://Anodya:Anodya123@cluster0.ta3qef8.mongodb.net/?retryWrites=true&w=m
        client = pymongo.MongoClient(atlas_connection_string)

        db = client["video_database"]
        videos_collection = db["videos"]

        # Check if the collection exists, and create it if it doesn't
        if "videos" not in db.list_collection_names():
            db.create_collection("videos")

        print("Connected to MongoDB video_database")

        return videos_collection

    except Exception as e:
        print("Error connecting to MongoDB:", e)
        return None

```

```python
        with open(video_path, "wb") as f:
            f.write(uploaded_file.getbuffer())

        # Process the video and make predictions
        frames = process_video(video_path, SEQUENCE_LENGTH)
        prediction = make_prediction(frames)

        # prediction_label = class_labels[prediction]
        # font_color = "red" if prediction_label == "Abnormal" else "blue"
        prediction_label = class_labels[prediction]
        font_color = "white"
        background_color = "red" if prediction_label == "Abnormal" else "blue"

        # prediction_text = f"<div style='text-align:center; font-size:24px; color:{font_color};'>{prediction_label}</div>"
        # st.markdown(prediction_text, unsafe_allow_html=True)

        prediction_text = (
            f"<div style='text-align:center; font-size:24px; "
            f"color:{font_color}; background-color:{background_color}; "
            f"padding:10px; border-radius:10px;'>{prediction_label}</div>"
        )
        st.markdown(prediction_text, unsafe_allow_html=True)



def process_video(video_path, sequence_length):
    # Rest of the function remains the same

    '''
    This function will extract the required frames from a video after resizing and normalizing them.
```

SLIIT FACULTY OF COMPUTING

Developed solution cont.



```python
        '<h2 style="color: darkblue;">Guidelines for Infant Video Recording</h2>',
        unsafe_allow_html=True
    )
    display_guidelines(guidelines)

# Initialize video-related variables
cap = None
video_placeholder = None
start_button = None
stop_button = None
is_recording = False
out = None
start_datetime = None
video_id = None
mp4_path = None  # Initialize the mp4_path variable

# Create different pages or sections
if page == "Record Video":
    st.markdown(
        '<h1 style="color: darkblue;">Baby Moments - Record a  Video</h1>',
        unsafe_allow_html=True
    )

    # Check if the image is clicked to initiate video recording
    if st.button("Click here to Record Video"):
        # Create a VideoCapture object to access the webcam
        cap = cv2.VideoCapture(0)

        # Create a placeholder to display the webcam feed
        video_placeholder = st.empty()

        # Create buttons to start and stop recording
        start_button = st.button("Start Recording")
```

Results and Discussion

Model Results and Accuracy of prediction

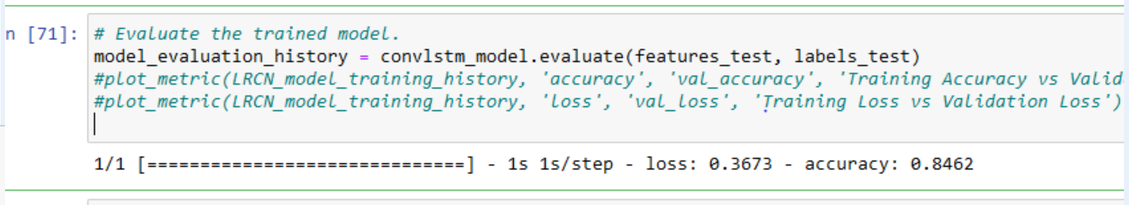
✓ Used two models

- **LRCN Model (Long-Short Term Memory Convolutional Neural Network)**
- **ConvLSTM Model (Convolutional Long-Short Term Memory Model)**

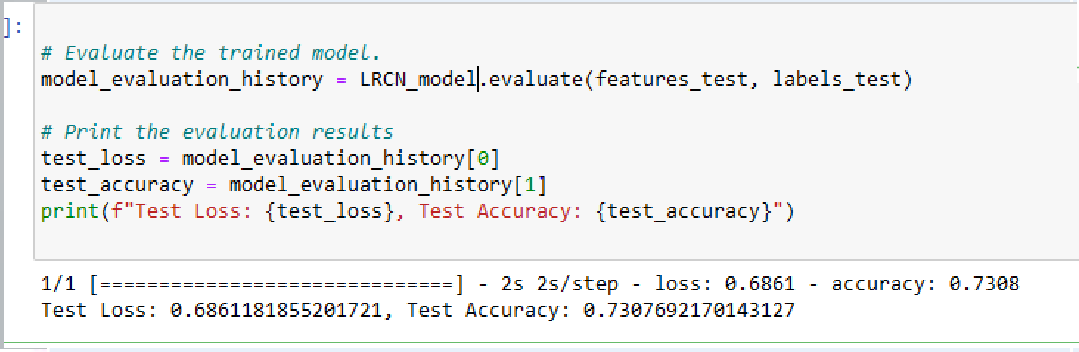
LRCN Model	ConvLSTM Model	Results
designed to analyze sequences of video frames and make predictions about infant behavior.	designed to analyze sequences of video frames and make predictions about infant behavior.	Same purpose
uses a combination of Convolutional Neural Networks (CNNs) to extract spatial features from individual frames and Long Short-Term Memory (LSTM) layers to capture temporal dependencies between frames in a sequence.	uses ConvLSTM layers, which combine the spatial feature extraction capability of CNNs with the sequential modeling ability of LSTMs.	Different Architectures

Results and Discussion cont.

LRCN Model	ConvLSTM Model	Results
takes a sequence of video frames as input, where each frame is resized, normalized, and processed.	it takes a sequence of video frames as input, which are resized and normalized.	Same input taken
The model is trained on a dataset of videos, with labels indicating whether the infant behavior is abnormal or normal. With Accuracy of : 1/1 [==============================] - 2s 2s/step - loss: 0.6861 - accuracy: 0.7308 Test Loss: 0.6861181855201721, Test Accuracy: 0.7307692170143127	The model is trained on a dataset of videos with behavior labels . With Accuracy of : 1/1 [==============================] - 1s 1s/step - loss: 0.3673 - accuracy: 0.8462	High accuracy rate in **ConvLSTM Model**



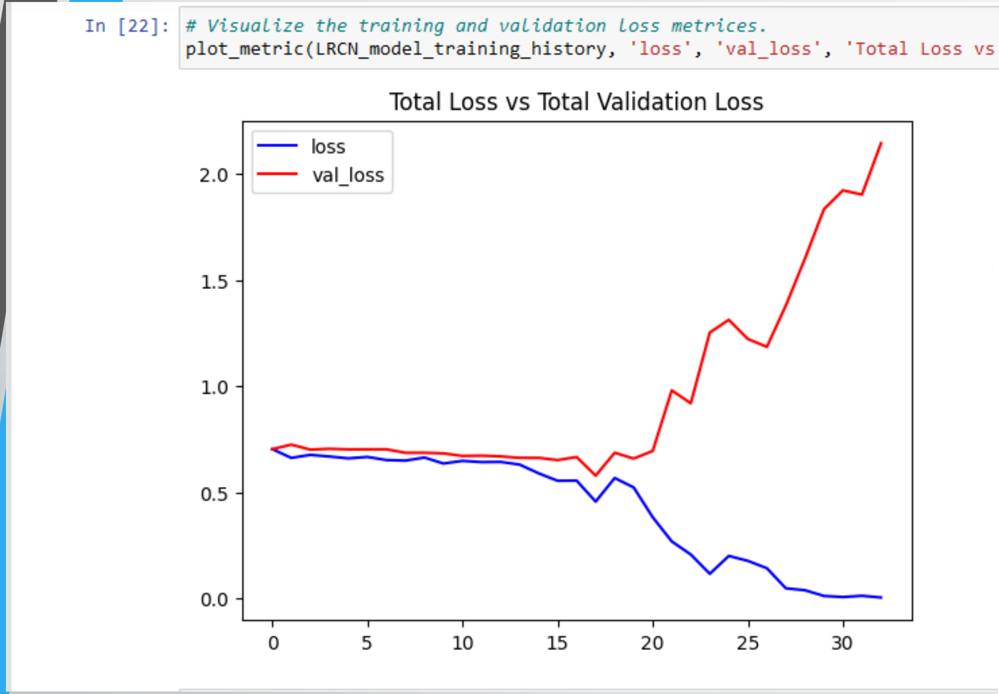
```
n [71]: # Evaluate the trained model.
        model_evaluation_history = convlstm_model.evaluate(features_test, labels_test)
        #plot_metric(LRCN_model_training_history, 'accuracy', 'val_accuracy', 'Training Accuracy vs Valid
        #plot_metric(LRCN_model_training_history, 'loss', 'val_loss', 'Training Loss vs Validation Loss')

        1/1 [==============================] - 1s 1s/step - loss: 0.3673 - accuracy: 0.8462
```



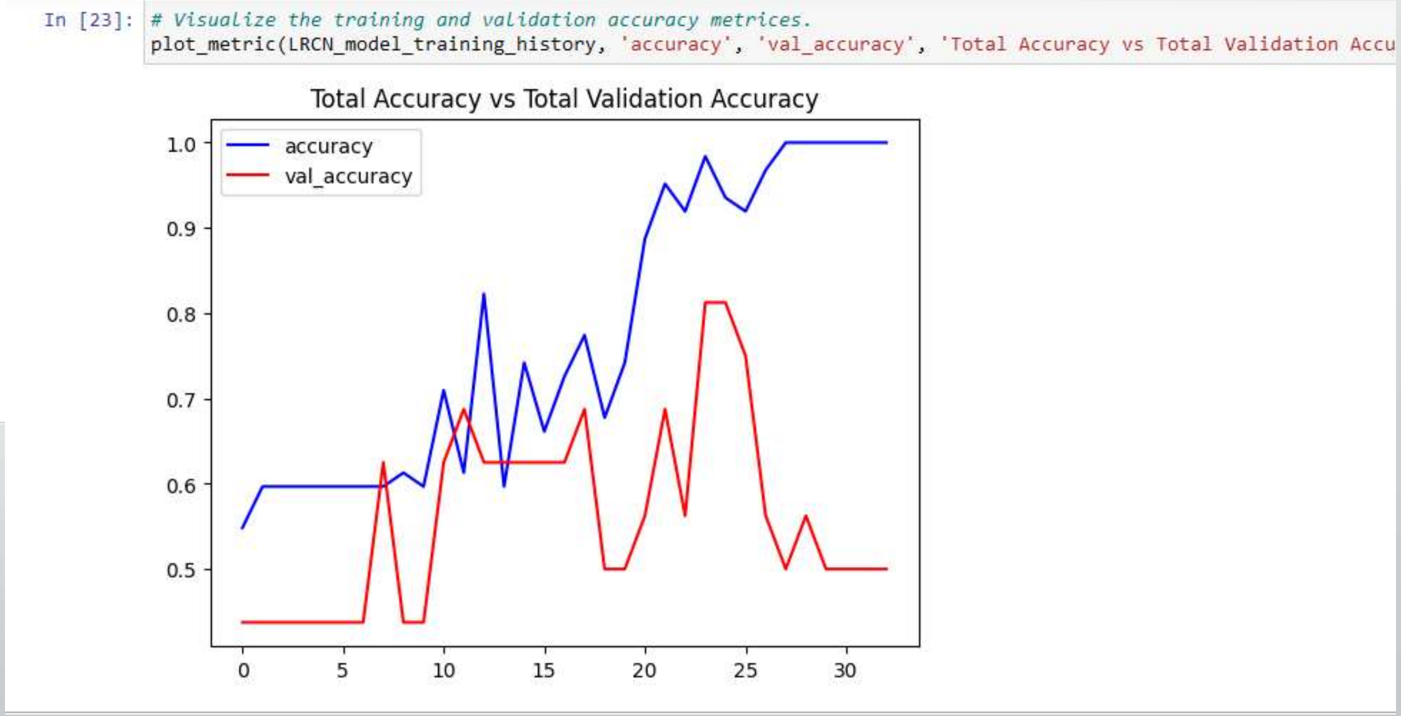
```
]:
    # Evaluate the trained model.
    model_evaluation_history = LRCN_model.evaluate(features_test, labels_test)

    # Print the evaluation results
    test_loss = model_evaluation_history[0]
    test_accuracy = model_evaluation_history[1]
    print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")

    1/1 [==============================] - 2s 2s/step - loss: 0.6861 - accuracy: 0.7308
    Test Loss: 0.6861181855201721, Test Accuracy: 0.7307692170143127
```

Results and Discussion cont.

- Due to its specialized layers, the ConvLSTM model is more complicated and may be able to detect more complex patterns in video sequences.



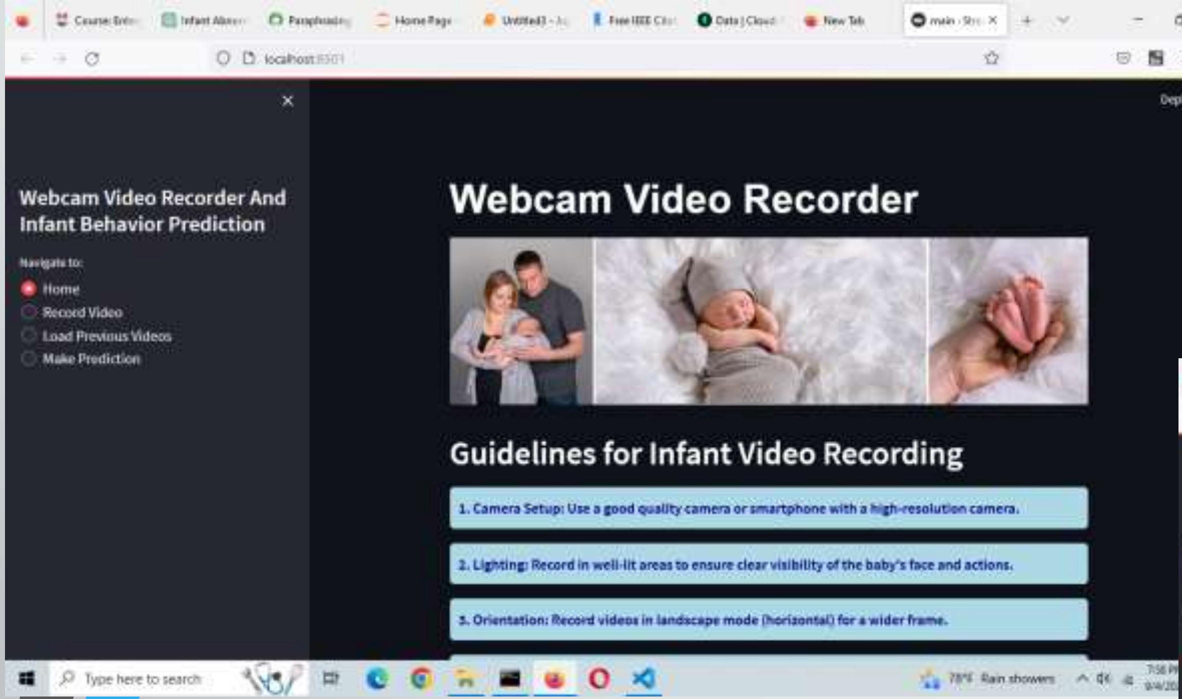
```
In [22]: # Visualize the training and validation loss metrices.
         plot_metric(LRCN_model_training_history, 'loss', 'val_loss', 'Total Loss vs Total Validation Loss')
```



```
In [23]: # Visualize the training and validation accuracy metrices.
         plot_metric(LRCN_model_training_history, 'accuracy', 'val_accuracy', 'Total Accuracy vs Total Validation Accu
```

Results and Discussion cont.

System Results and prediction outputs



Webcam Video Recorder And Infant Behavior Prediction

Navigate to:

- ● Home
- ○ Record Video
- ○ Load Previous Videos
- ○ Make Prediction

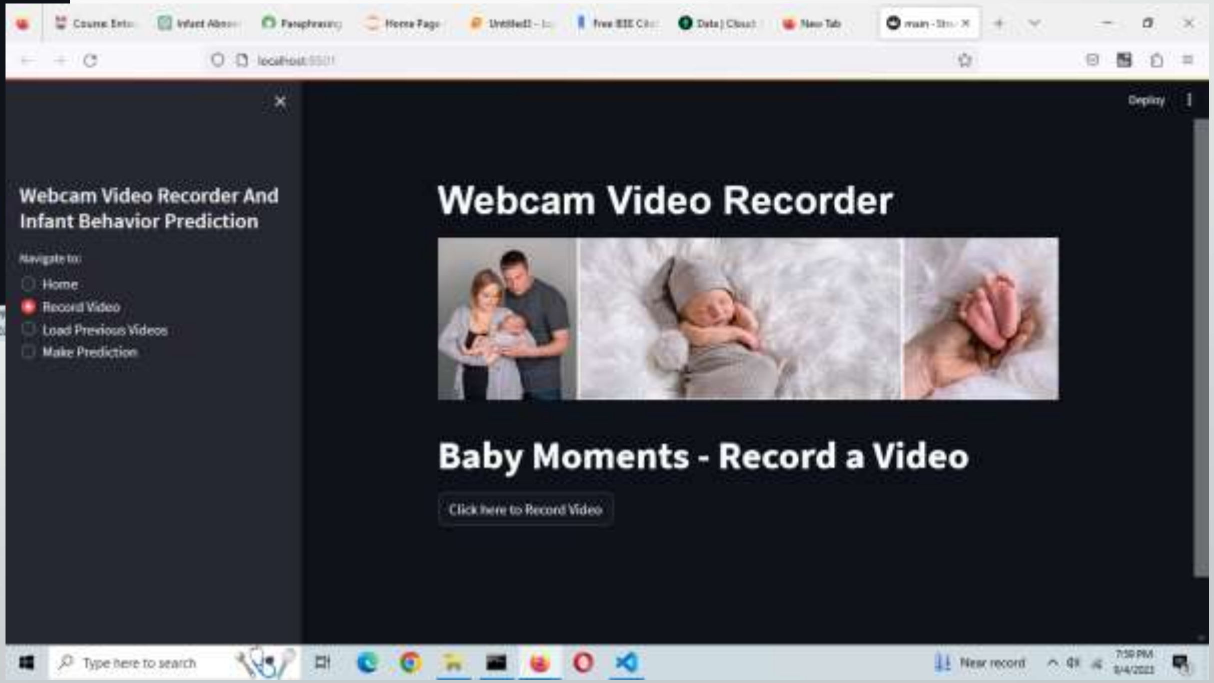
Webcam Video Recorder

Guidelines for Infant Video Recording

1. Camera Setup: Use a good quality camera or smartphone with a high-resolution camera.

2. Lighting: Record in well-lit areas to ensure clear visibility of the baby's face and actions.

3. Orientation: Record videos in landscape mode (horizontal) for a wider frame.



Webcam Video Recorder And Infant Behavior Prediction

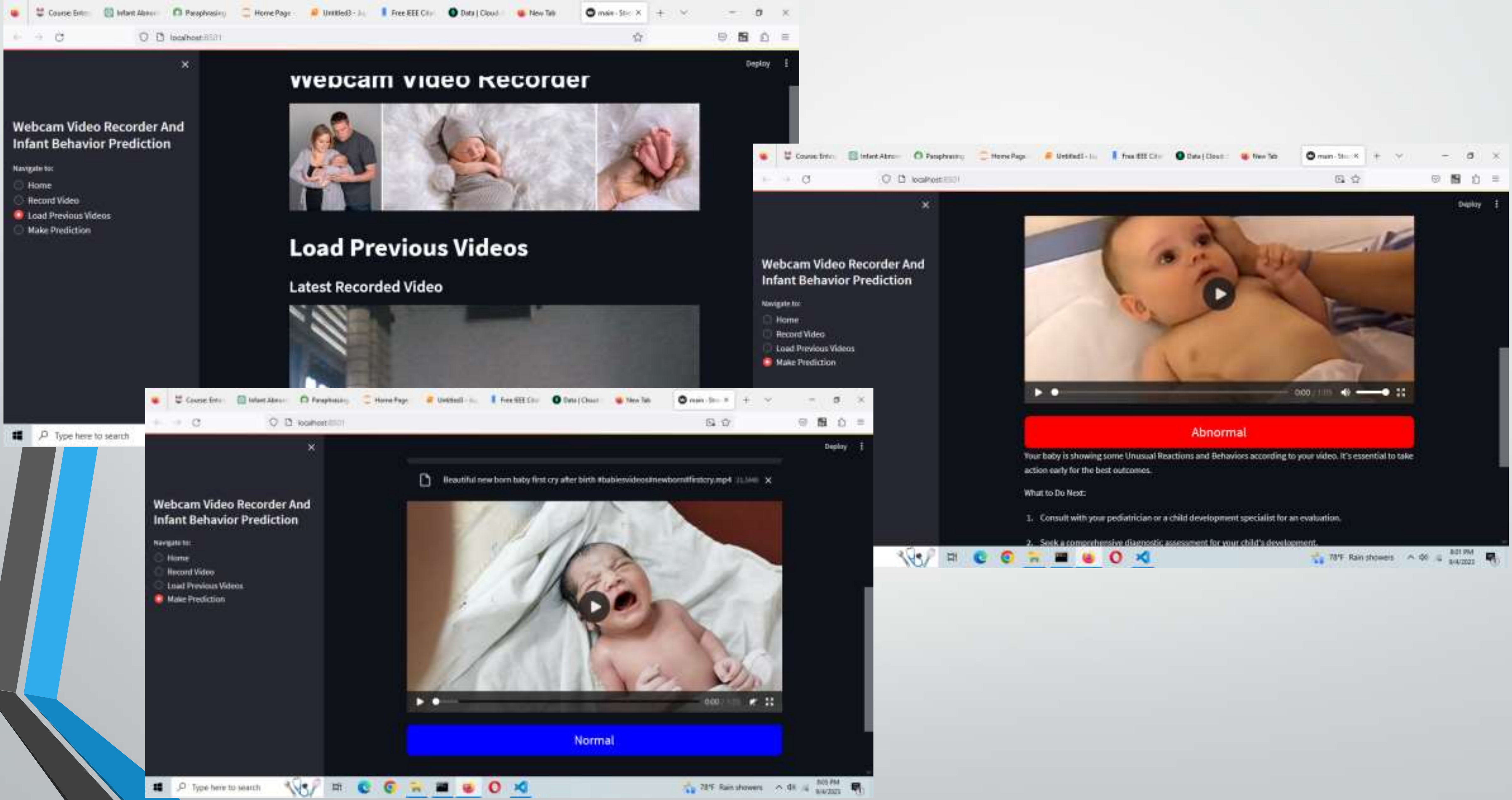
Navigate to:

- ○ Home
- ● Record Video
- ○ Load Previous Videos
- ○ Make Prediction

Webcam Video Recorder

Baby Moments - Record a Video

Click here to Record Video



Load Previous Videos

Latest Recorded Video

Webcam Video Recorder And Infant Behavior Prediction
Navigate to:
○ Home
○ Record Video
○ Load Previous Videos
● Make Prediction
Beautiful new born baby first cry after birth #babiesvideos#newbornfirstcry.mp4
Normal

Your baby is showing some Unusual Reactions and Behaviors according to your video. It's essential to take action early for the best outcomes.

What to Do Next:

1. Consult with your pediatrician or a child development specialist for an evaluation.
2. Seek a comprehensive diagnostic assessment for your child's development.

References

[1] B. Z. Karmel *et al.*, “Early Medical and Behavioral Characteristics of NICU Infants Later Classified With ASD,” *Pediatrics*, vol. 126, no. 3, pp. 457–467, Sep. 2010, doi: https://doi.org/10.1542/peds.2009-2680.

[2]T. E. Inder, “Abnormal Cerebral Structure Is Present at Term in Premature Infants,” *PEDIATRICS*, vol. 115, no. 2, pp. 286–294, Feb. 2005, doi: https://doi.org/10.1542/peds.2004-0326.

[3]A. J. Spittle *et al.*, “Early emergence of behavior and social-emotional problems in very preterm infants,” *Journal of the American Academy of Child and Adolescent Psychiatry*, vol. 48, no. 9, pp. 909–918, Sep. 2009, doi: https://doi.org/10.1097/CHI.0b013e3181af8235.

[4]M. Hadders-Algra, “Evaluation of Motor Function in Young Infants by Means of the Assessment of General Movements: A Review,” *Pediatric Physical Therapy*, vol. 13, no. 1, p. 27, 2001, Accessed: Sep. 04, 2023. [Online]. Available: https://journals.lww.com/pedpt/Fulltext/2001/04000/Evaluation_of_Motor_Function_in_Young_Infants_by.5.aspx

Thank You!

Team TMP 23-103