Major project final defense
on

Virtual Fitness Coach

Under the Supervision of
Rishi K. Marseni

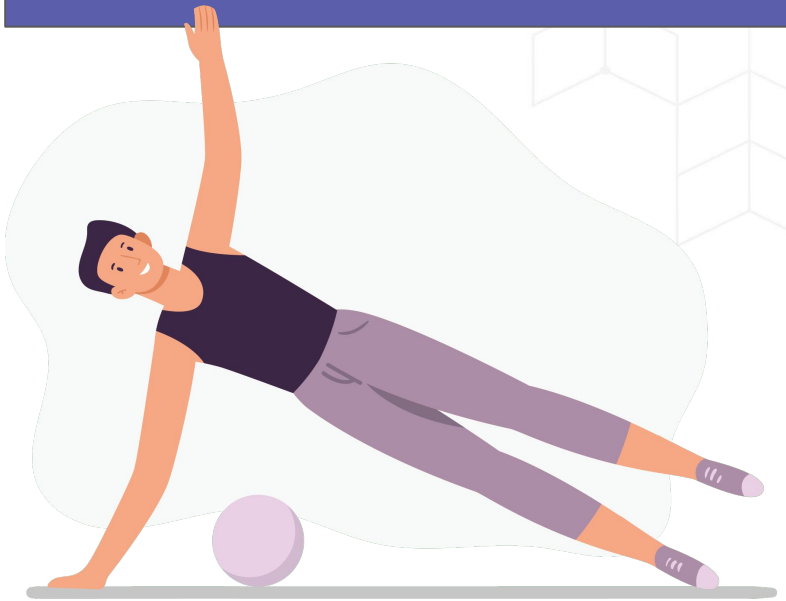
Team Members

Asbin Khanal, 181507
Aayush Raj Regmi, 181509
Manita Kunwor, 181523
Rohil Maharjan, 181535
Thomas Basyal, 181546

Department of IT Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal



Transforming Fitness,
One Virtual Step at a Time

Introduction

- VFC: innovative technology in this genre
- reduce shortcomings of traditional fitness systems
- Employs machine learning for health care
- Provides a convenient and effective approach for individuals
- To attain their fitness goals without relying on a physical trainer.
- Analyzes the user's fitness data and provides a customized plans
- The system continually learns from the user's data

Problem Statement

- One-size-fits-all approach in traditional fitness
- Divided Attention
- Specific location for fitness, so user inconvenience
- Generalized fitness goals and improper techniques
- False Assumptions regarding diet.
- Does not take into account too much of user’s data

Project Objective

- Provide personalized diet recommendations
- Provide workout guide
- Personalized fitness coaching based on user goals.
- Real time feedback and monitoring using posture detection.
- Adaptive coaching based on user data for optimal results.
- Customer support chat system

Project Scope

- To provide customized diet plans according to user’s information
- Use Computer vision for monitoring and real-time feedback.
- To provide customized workout plan and proper guidance
- Chatbot for customer support and basic guidance
- Mobile and web application development
- User’s dashboard for proper goal tracking
- Provide user notifications in regular time interval.
- Homepage where experts i.e. Doctors and Trainers can push contents regarding healthy habits.

Limitations

- Dependency on user’s data for personalized coaching.
- Lack of technological infrastructure
- May not account complex postures and detail muscles
- May not include cultural variations in diet plan
- Accuracy might vary

Related Works

- Diet recommendations according to BMI information and using K-means
- Body Tracking using Mediapipe and MoveNet.
- Knowledge based conversational AI tool : GODEL (Grounded Open Dialog Language Model)
- Fitness Blog.
- A Food Recommender System Considering Nutritional Information and User Preferences

Software Process Model

- Suitable for lengthy development schedules, new technology, and high-risk projects.
- Major requirements defined, some details evolve with time.
- Divide and conquer approach for better management.
- Lower initial delivery costs.
- Core modules are tested from the beginning - easy to detect errors.
- Generates working software quickly- allows early feedback and testing.

Software Process Model

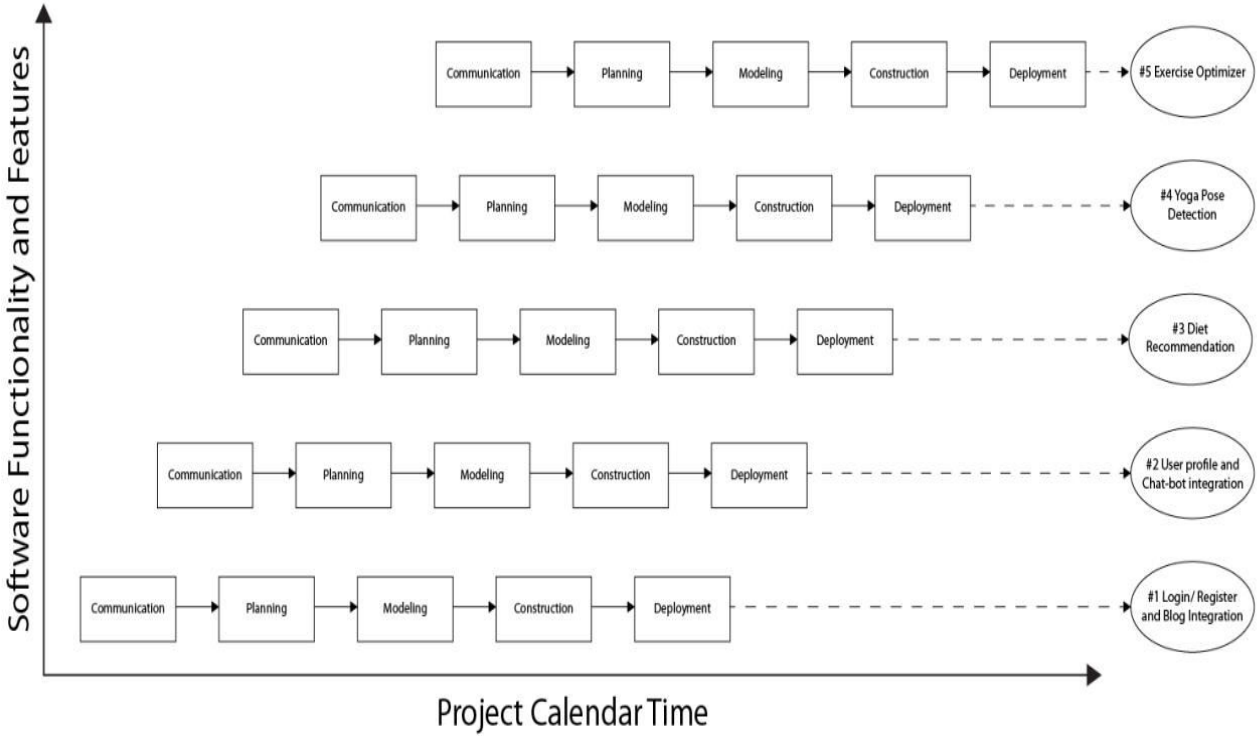
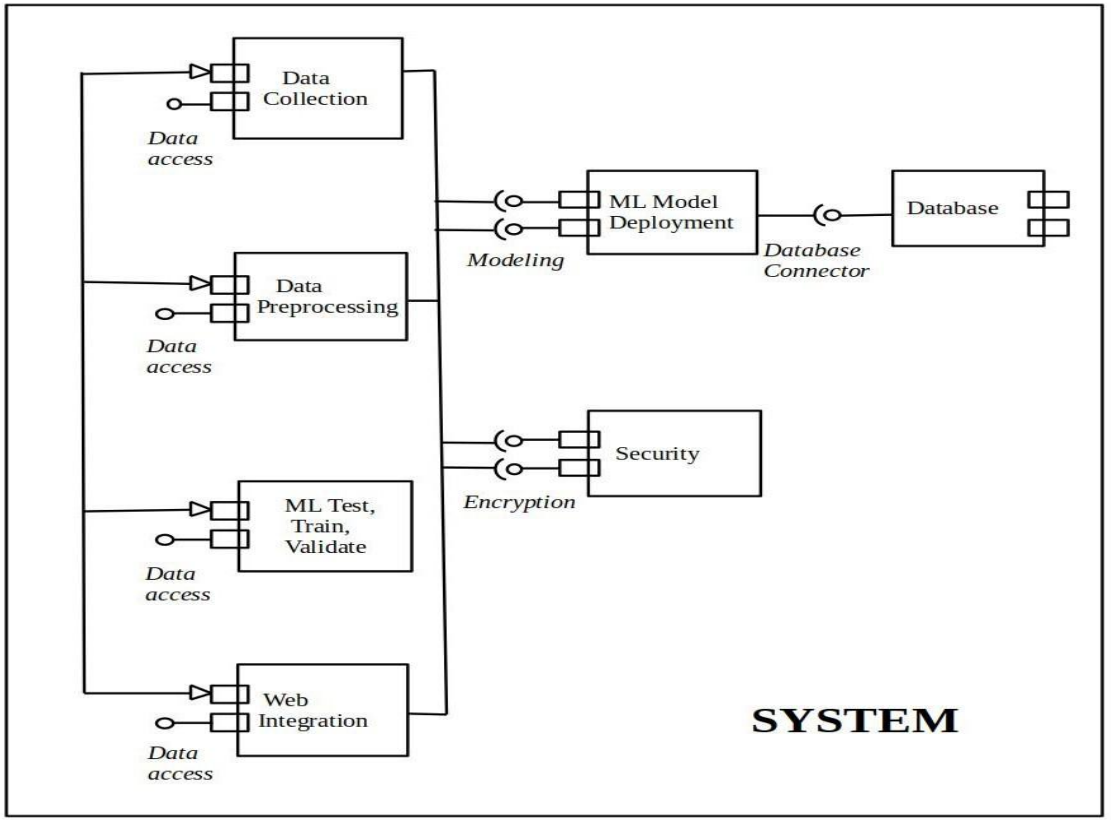
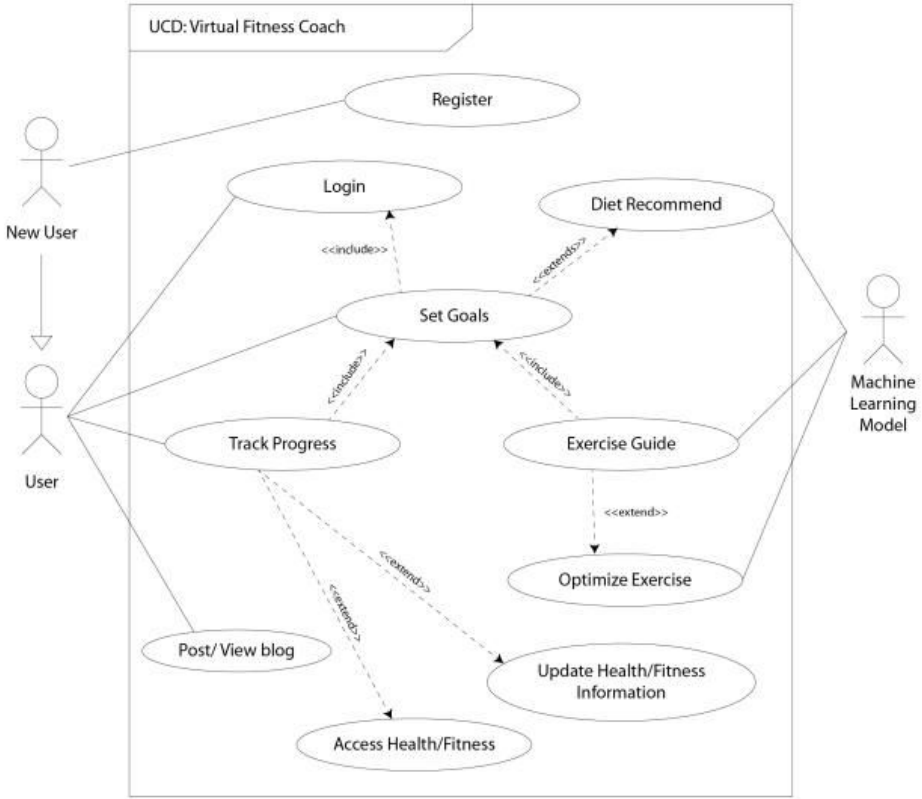


Figure: Incremental Model

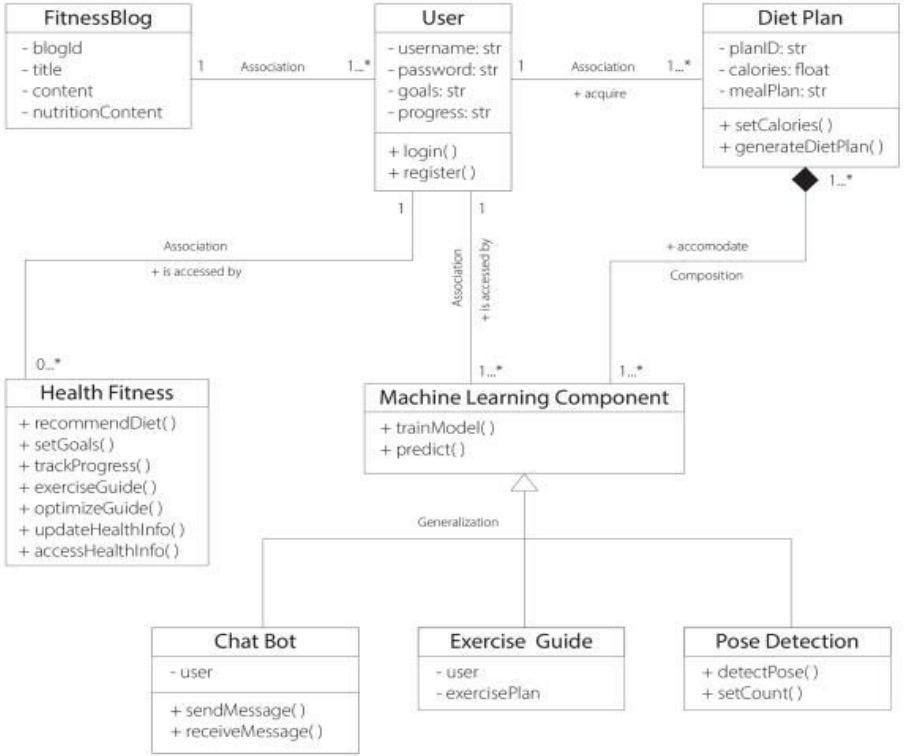
Virtual Fitness Coach System Architecture



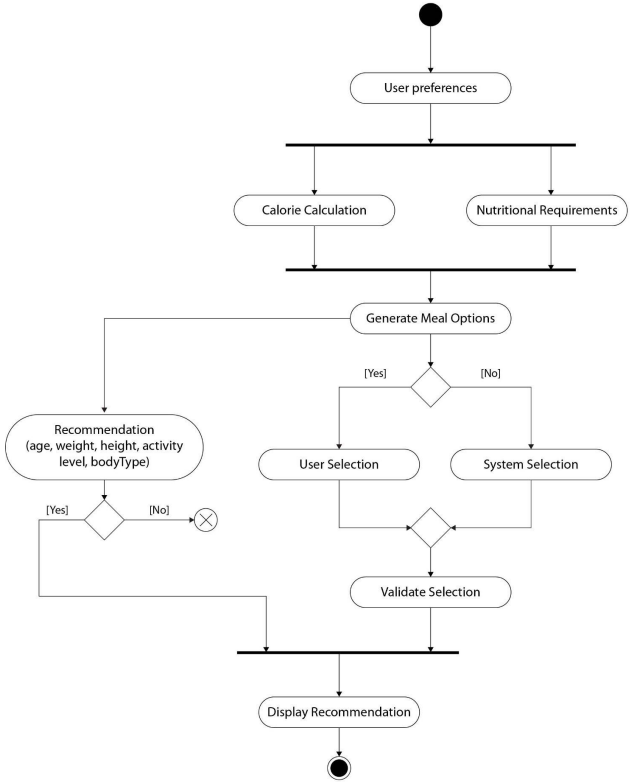
Virtual Fitness Coach Use Case Diagram



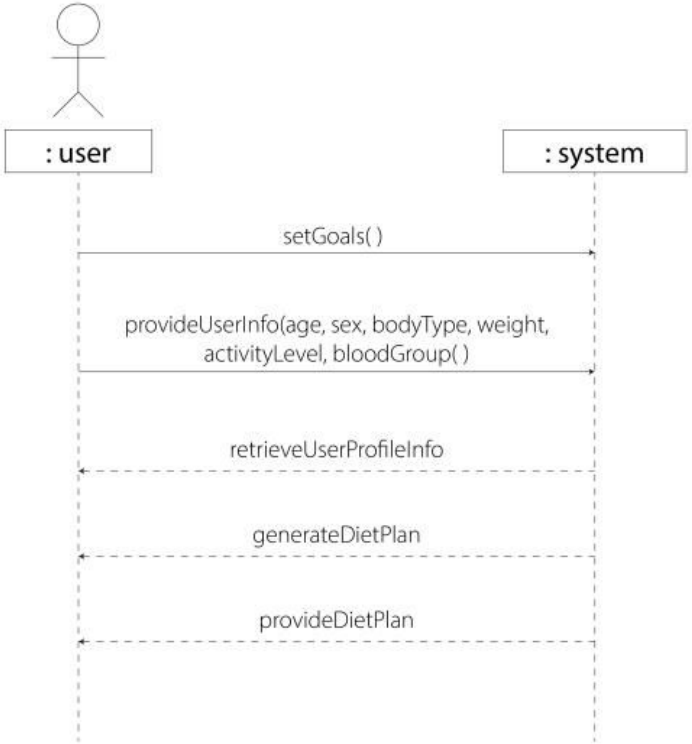
Virtual Fitness Coach Class Diagram



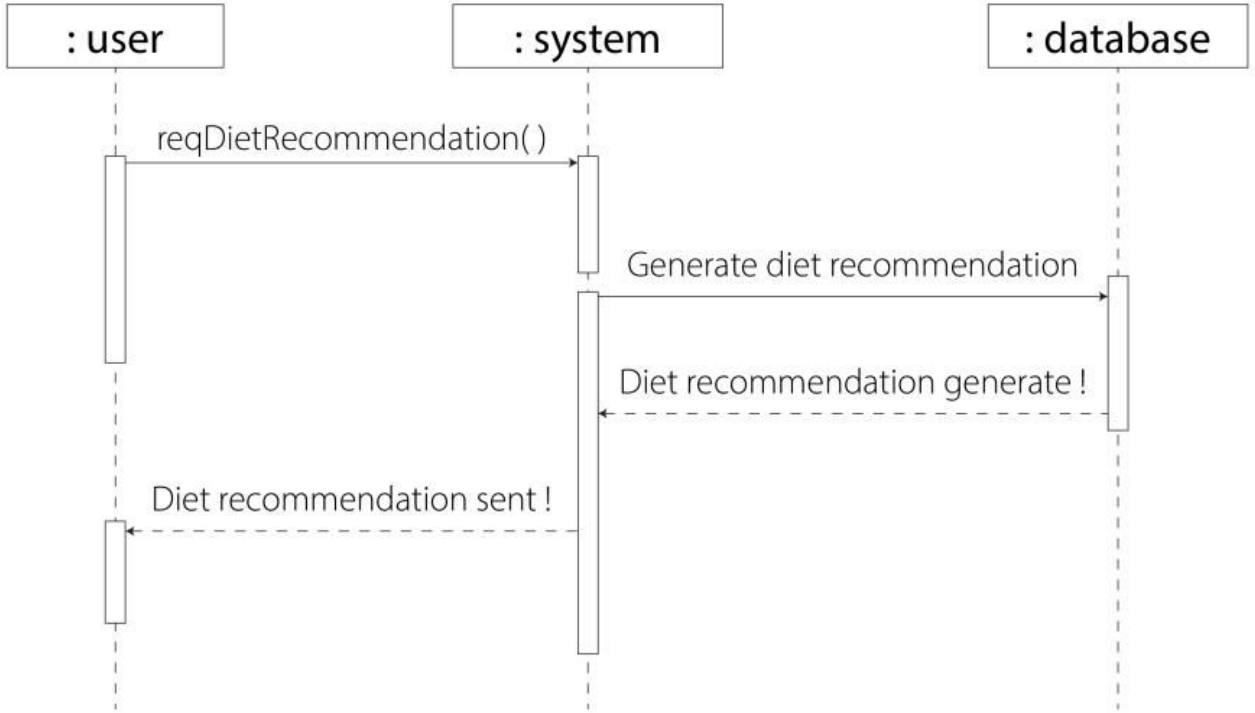
Activity Diagram for Diet Recommendation



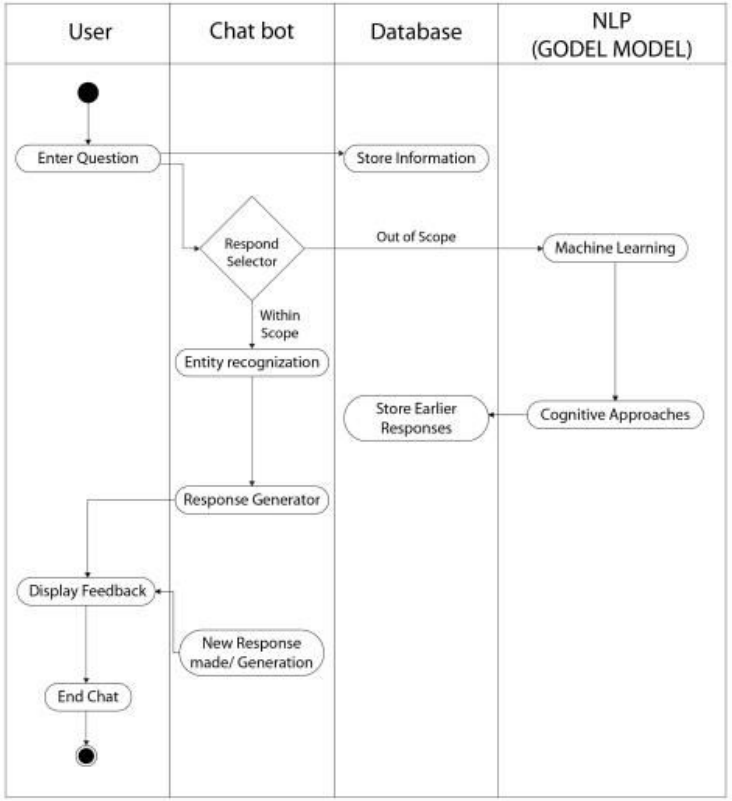
System Sequence of Diet Recommendation



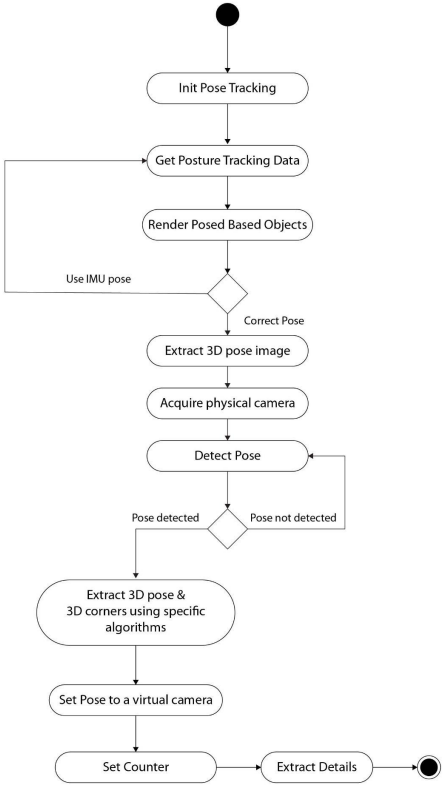
Sequence of Diet Recommendation



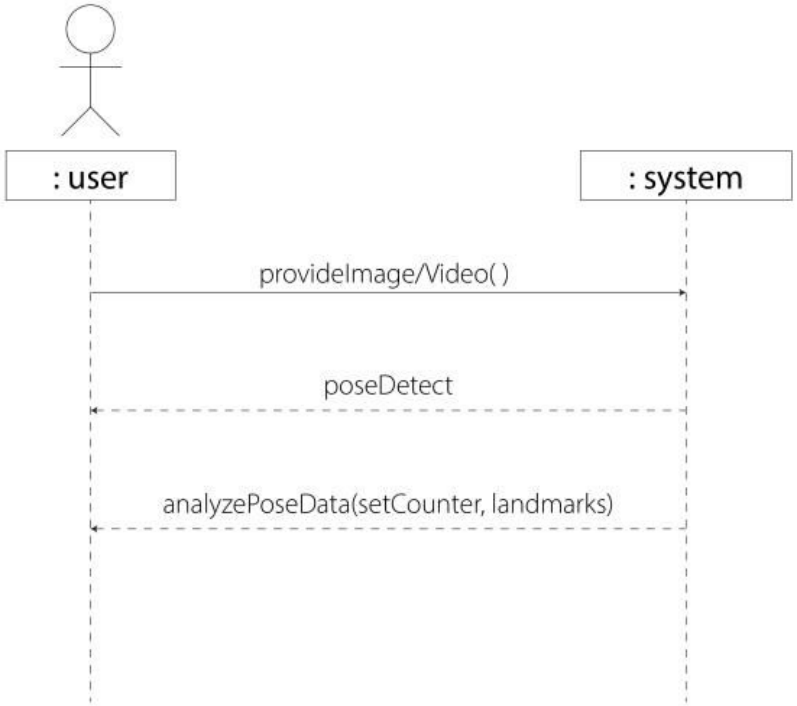
Activity Diagram for Chatbot



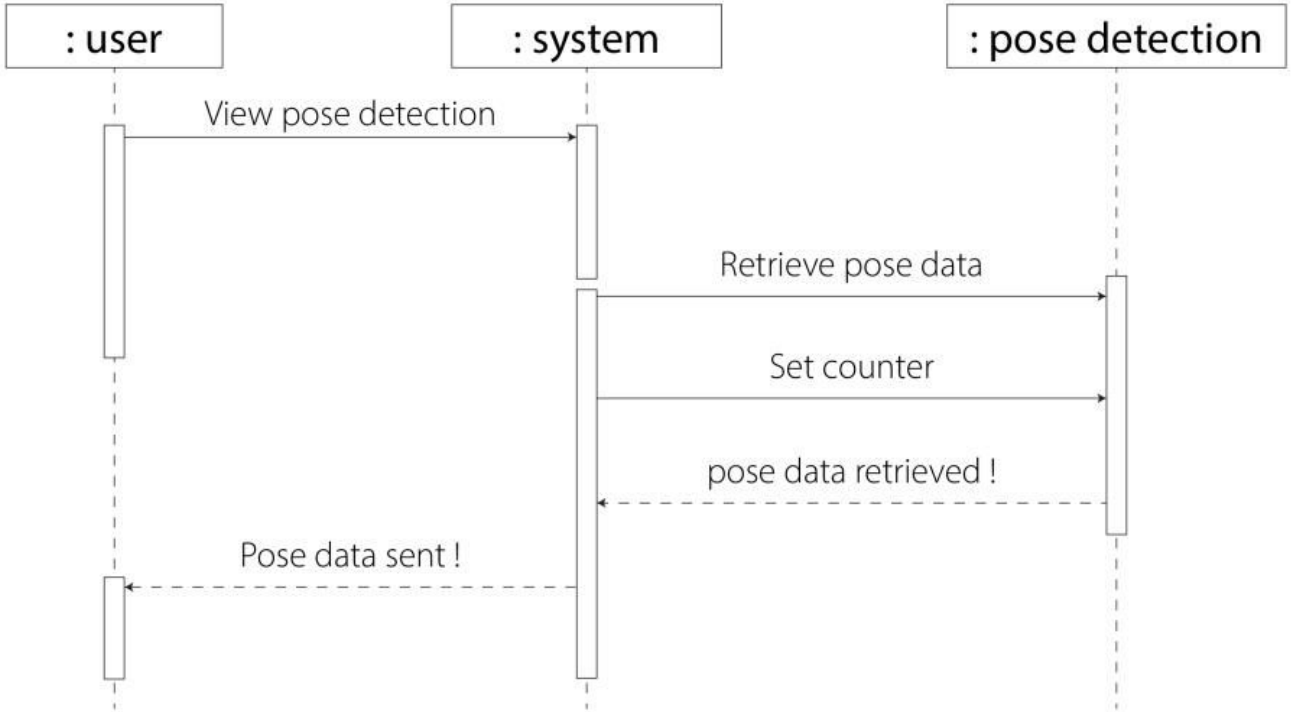
Activity Diagram for Pose Detection



System Sequence of Pose Detection



Sequence of Pose Detection



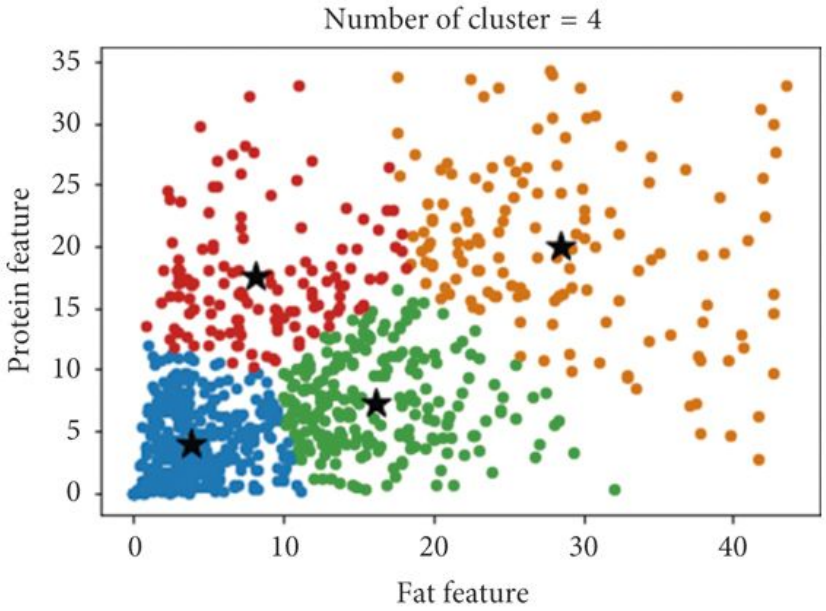
Diet Recommendation

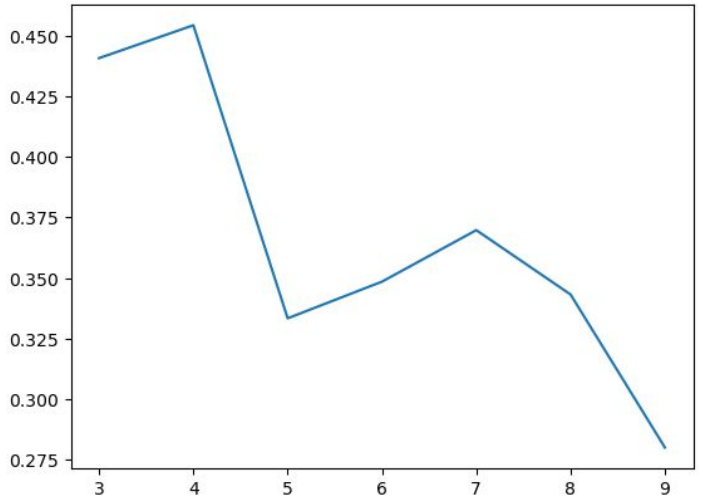
Dataset:

- https://github.com/vishalvermaCred/ML_project/blob/master/food.csv

1	Food_Items	Breakfast	Lunch	Dinner	VegNovVeg	Calories	Fats	Proteins	Iron	Calcium	Sodium	Potassium	Carbohydrates	Fibre	VitaminD	Sugars
2	Asparagus Cooked	0	1	1		22	0.2	2.4	0.91	23	14	224	4.1	2	0	1.3
3	Avocados	1	0	0	0	160	15	2	0.55	12	7	485	8.5	6.7	0	0.7
4	Bananas	1	0	0	0	89	0.3	1.1	0.26	5	1	358	23	2.6	0	12
5	Bagels made in wheat	0	1	1	0	250	1.5	10	2.76	20	439	165	49	4.1	0	6.1
6	Berries	1	0	0	0	349	0.4	14	6.8	190	298	77	77	13	0	46
7	Brocolli	0	1	1	0	25	0.5	3.8	1.27	118	56	343	3.1	2.8	0	0.6
8	Brown Rice	0	1	1	0	362	2.7	7.5	1.8	33	4	268	76	3.4	0	0

Diet Recommendation



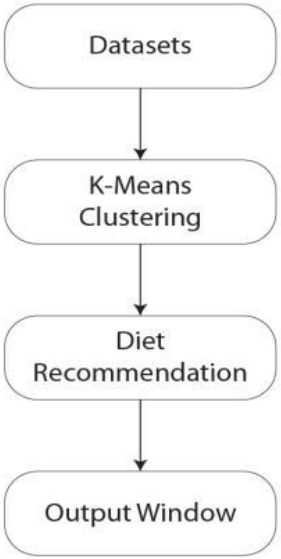


Silhouette score for Hyperparameter Tuning

Diet Recommendation

- In order to assign the users to the food clusters based on their body information, we utilized external validation methods.

- Similarly, the foods were recommended.



Diet Recommendation

Datasets

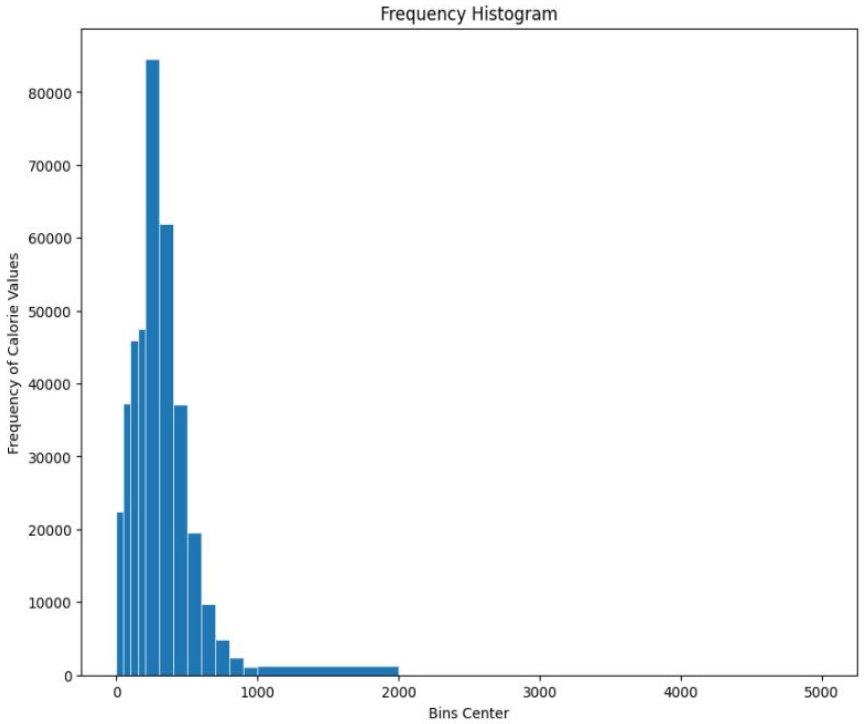
- https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews?select=recipes.csv

- Contains 522,517 recipes from 312 different categories
- Provides information about each recipe like cooking times, servings, ingredients, nutrition, instructions, and more.

Diet Dataset

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 375703 entries, 0 to 375702
Data columns (total 16 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   RecipeId              375703 non-null  int64
 1   Name                  375703 non-null  object
 2   CookTime              375703 non-null  int64
 3   PrepTime              375703 non-null  int64
 4   TotalTime             375703 non-null  int64
 5   RecipeIngredientParts 375703 non-null  object
 6   Calories              375703 non-null  float64
 7   FatContent            375703 non-null  float64
 8   SaturatedFatContent   375703 non-null  float64
 9   CholesterolContent    375703 non-null  float64
 10  SodiumContent         375703 non-null  float64
 11  CarbohydrateContent   375703 non-null  float64
 12  FiberContent          375703 non-null  float64
 13  SugarContent          375703 non-null  float64
 14  ProteinContent        375703 non-null  float64
 15  RecipeInstructions    375703 non-null  object
dtypes: float64(9), int64(4), object(3)
memory usage: 45.9+ MB
```



Model Fitting

Trainning the model

Fitting the model

In [24]:
```
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(metric='cosine',algorithm='brute')
neigh.fit(prep_data)
```

Out[24]:
```
NearestNeighbors
NearestNeighbors(algorithm='brute', metric='cosine')
```

In [25]:
```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import FunctionTransformer
transformer = FunctionTransformer(neigh.kneighbors,kw_args={'return_distance':False})
pipeline=Pipeline([('std_scaler',scaler),('NN',transformer)])
```

In [26]:
```
params={'n_neighbors':10,'return_distance':False}
pipeline.get_params()
pipeline.set_params(NN__kw_args=params)
```

Out[26]:
```
Pipeline
StandardScaler
FunctionTransformer
```

API Output Format

http://127.0.0.1:8000/predict/

Server response

Code	Details
200	Response body


```
{
  "output": [
    {
      "Name": "Smoked Salmon, Cucumber &amp;amp; Horseradish Lavash Sandwiches",
      "CookTime": "0",
      "PrepTime": "40",
      "TotalTime": "40",
      "RecipeIngredientParts": [
        "English cucumber",
        "red onion",
        "sour cream",
        "cream cheese",
        "capers",
        "smoked salmon",
        "fresh chives",
        "watercress leaves"
      ],
      "Calories": 329.2,
      "FatContent": 16.4,
      "SaturatedFatContent": 7.9,
      "CholesterolContent": 38.4,
      "SodiumContent": 482.5,
      "CarbohydrateContent": 28.3,
      "FiberContent": 14,
      "SugarContent": 11.6,
      "ProteinContent": 27.2,
      "RecipeInstructions": [
        "Mix sour cream, cream cheese, capers, horseradish and salt and pepper.",
        "Arrange all ingredients so that you have two equal amounts (for the two rolls).",
```

Pose Tracking and Feedback

Datasets were taken from these sources

https://www.kaggle.com/datasets/muhannadtuameh/exercise-recognition

- Contains 10 different physical poses
- Distinguished in 5 exercises
- Push-up, Pull-up, Sit-up, Jumping Jack and Squat.
- For every exercise, 2 different classes have been used to represent the terminal positions of that exercise (e.g., “up” and “down” positions for push-ups).

Pose Tracking and Feedback

http://download.tensorflow.org/data/pose_classification/yoga_poses.zip

- Contains data of classes:
  - Chair
  - Cobra
  - Dog
  - Tree
  - Warrior
  - Triangle
  - Shoulder Stand

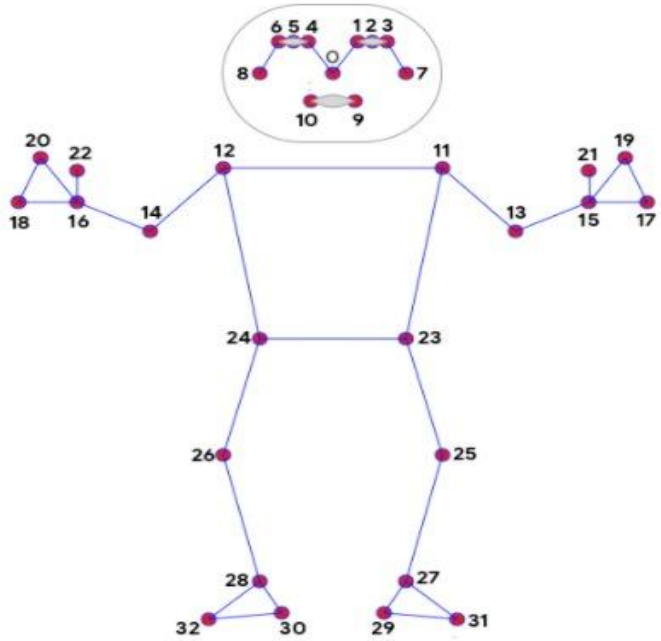
```
In [17]: print("Chair Data: ",chair.shape)
         print("Cobra Data: ",cobra.shape)
         print("Dog Data: ",dog.shape)
         print("Shoulder Stand Data: ",shoulder_stand.shape)
         print("Triangle Data: ",triangle.shape)
         print("Tree Data: ",tree.shape)
         print("Warrior Data: ",warrior.shape)

Chair Data:  (167, 52)
Cobra Data:  (193, 52)
Dog Data:  (168, 52)
Shoulder Stand Data:  (7, 52)
Triangle Data:  (9, 52)
Tree Data:  (186, 52)
Warrior Data:  (138, 52)
```

Also various data augmentation measures are performed

Pose Tracking and Feedback

Pose Landmark Model (BlazePose GHUM 3D)

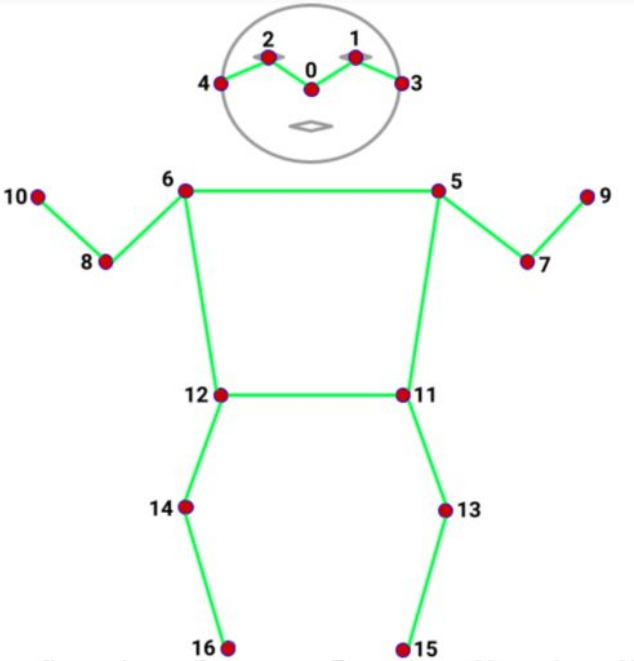
The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks (see figure below).



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Fig 4. 33 pose landmarks.

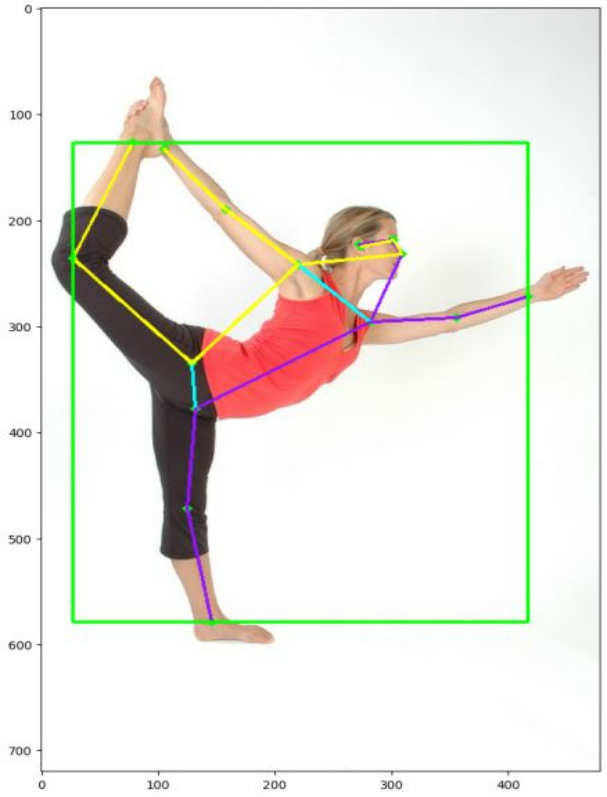
Pose Tracking and Feedback




```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1512 entries, 0 to 1511
Data columns (total 54 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   filename              1512 non-null   object
 1   NOSE_x                1512 non-null   float64
 2   NOSE_y                1512 non-null   float64
 3   NOSE_score            1512 non-null   float64
 4   LEFT_EYE_x            1512 non-null   float64
 5   LEFT_EYE_y            1512 non-null   float64
 6   LEFT_EYE_score        1512 non-null   float64
 7   RIGHT_EYE_x           1512 non-null   float64
 8   RIGHT_EYE_y           1512 non-null   float64
 9   RIGHT_EYE_score       1512 non-null   float64
 10  LEFT_EAR_x            1512 non-null   float64
 11  LEFT_EAR_y            1512 non-null   float64
 12  LEFT_EAR_score        1512 non-null   float64
 13  RIGHT_EAR_x           1512 non-null   float64
 14  RIGHT_EAR_y           1512 non-null   float64
 15  RIGHT_EAR_score       1512 non-null   float64
 16  LEFT_SHOULDER_x       1512 non-null   float64
 17  LEFT_SHOULDER_y       1512 non-null   float64
 18  LEFT_SHOULDER_score   1512 non-null   float64
 19  RIGHT_SHOULDER_x      1512 non-null   float64
 20  RIGHT_SHOULDER_y      1512 non-null   float64
```

Pose Tracking and Feedback




```
 20  RIGHT_SHOULDER_y      1512 non-null    float64
 21  RIGHT_SHOULDER_score  1512 non-null    float64
 22  LEFT_ELBOW_x          1512 non-null    float64
 23  LEFT_ELBOW_y          1512 non-null    float64
 24  LEFT_ELBOW_score      1512 non-null    float64
 25  RIGHT_ELBOW_x         1512 non-null    float64
 26  RIGHT_ELBOW_y         1512 non-null    float64
 27  RIGHT_ELBOW_score     1512 non-null    float64
 28  LEFT_WRIST_x          1512 non-null    float64
 29  LEFT_WRIST_y          1512 non-null    float64
 30  LEFT_WRIST_score      1512 non-null    float64
 31  RIGHT_WRIST_x         1512 non-null    float64
 32  RIGHT_WRIST_y         1512 non-null    float64
 33  RIGHT_WRIST_score     1512 non-null    float64
 34  LEFT_HIP_x            1512 non-null    float64
 35  LEFT_HIP_y            1512 non-null    float64
 36  LEFT_HIP_score        1512 non-null    float64
 37  RIGHT_HIP_x           1512 non-null    float64
 38  RIGHT_HIP_y           1512 non-null    float64
 39  RIGHT_HIP_score       1512 non-null    float64
 40  LEFT_KNEE_x           1512 non-null    float64
 41  LEFT_KNEE_y           1512 non-null    float64
 42  LEFT_KNEE_score       1512 non-null    float64
 43  RIGHT_KNEE_x          1512 non-null    float64
 44  RIGHT_KNEE_y          1512 non-null    float64
 45  RIGHT_KNEE_score      1512 non-null    float64
 46  LEFT_ANKLE_x          1512 non-null    float64
 47  LEFT_ANKLE_y          1512 non-null    float64
 48  LEFT_ANKLE_score      1512 non-null    float64
 49  RIGHT_ANKLE_x         1512 non-null    float64
 50  RIGHT_ANKLE_y         1512 non-null    float64
 51  RIGHT_ANKLE_score     1512 non-null    float64
 52  class_no              1512 non-null    int64
 53  class_name            1512 non-null    object
dtypes: float64(51), int64(1), object(2)
memory usage: 638.0+ KB
```

Model Training

Our Keras model takes the detected pose landmarks, then calculates the pose embedding and predicts the pose class.

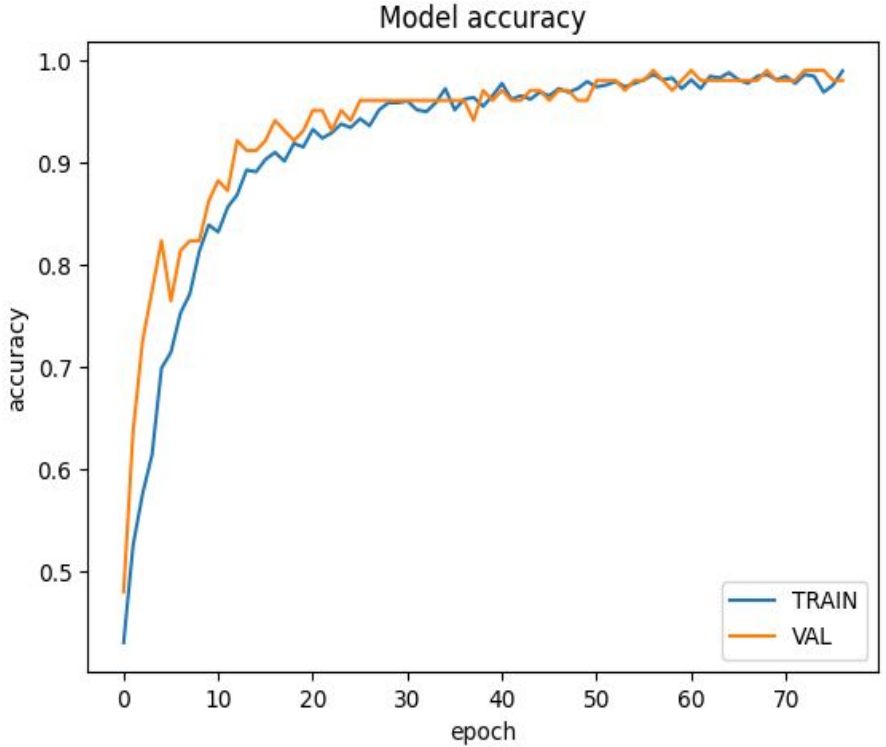
In [31]:
```
# Define the model
inputs = tf.keras.Input(shape=(51))
embedding = landmarks_to_embedding(inputs)

layer = keras.layers.Dense(128, activation=tf.nn.relu6)(embedding)
layer = keras.layers.Dropout(0.5)(layer)
layer = keras.layers.Dense(64, activation=tf.nn.relu6)(layer)
layer = keras.layers.Dropout(0.5)(layer)
outputs = keras.layers.Dense(len(class_names), activation="softmax")(layer)

model = keras.Model(inputs, outputs)
model.summary()
```

```
                                                                 'tf.math.maximum[0][0]']

 flatten (Flatten)              (None, 34)           0           ['tf.math.truediv[0][0]']

 dense (Dense)                  (None, 128)          4480        ['flatten[0][0]']

 dropout (Dropout)              (None, 128)          0           ['dense[0][0]']

 dense_1 (Dense)                (None, 64)           8256        ['dropout[0][0]']

 dropout_1 (Dropout)            (None, 64)           0           ['dense_1[0][0]']

 dense_2 (Dense)                (None, 5)            325         ['dropout_1[0][0]']

==================================================================================================
Total params: 13,061
Trainable params: 13,061
Non-trainable params: 0
__________________________________________________________________________________________________
```

Model Training

In [32]:
```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

# Add a checkpoint callback to store the checkpoint that has the highest
# validation accuracy.
checkpoint_path = "weights.best.hdf5"
checkpoint = keras.callbacks.ModelCheckpoint(checkpoint_path,
                             monitor='val_accuracy',
                             verbose=1,
                             save_best_only=True,
                             mode='max')
earlystopping = keras.callbacks.EarlyStopping(monitor='val_accuracy',
                                              patience=20)

# Start training
history = model.fit(X_train, y_train,
                    epochs=200,
                    batch_size=16,
                    validation_data=(X_val, y_val),
                    callbacks=[checkpoint, earlystopping])
```

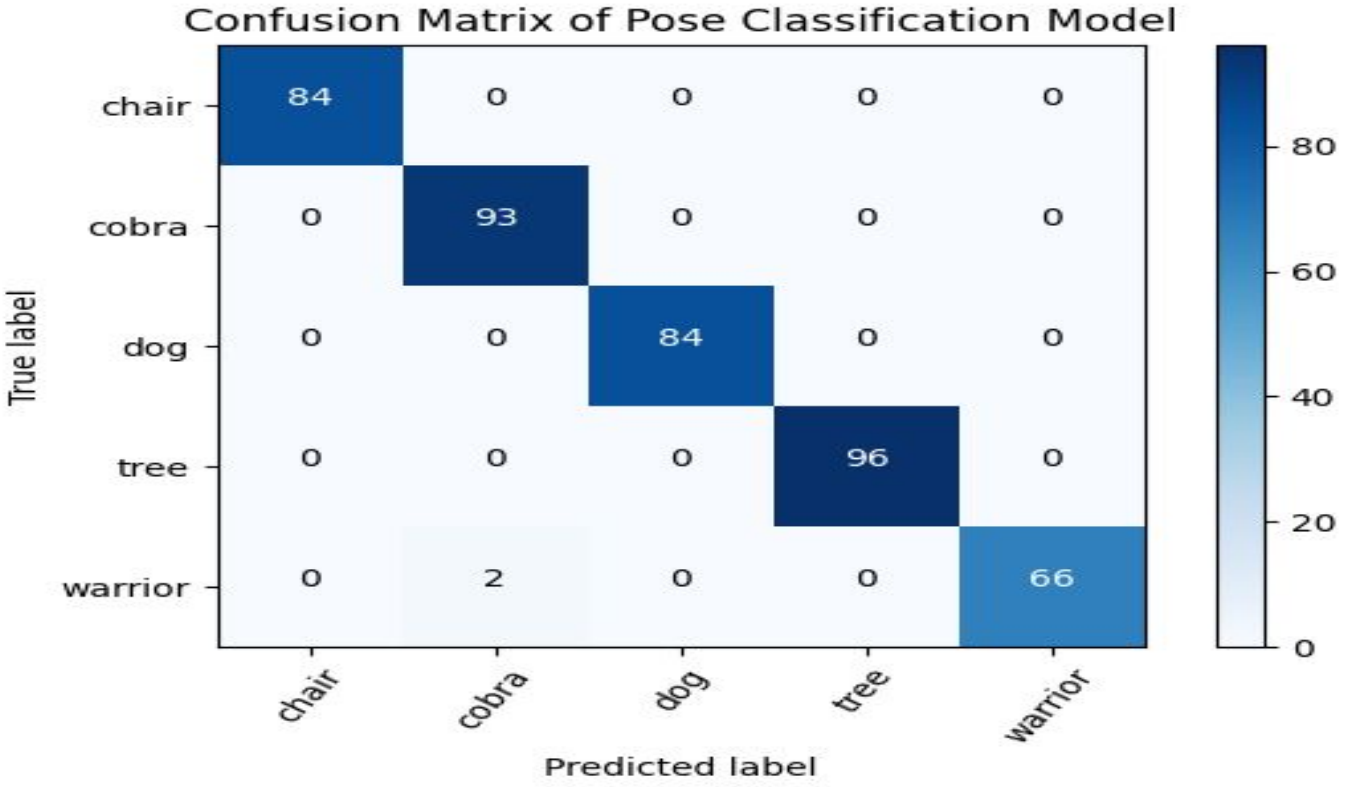
Model Evaluation



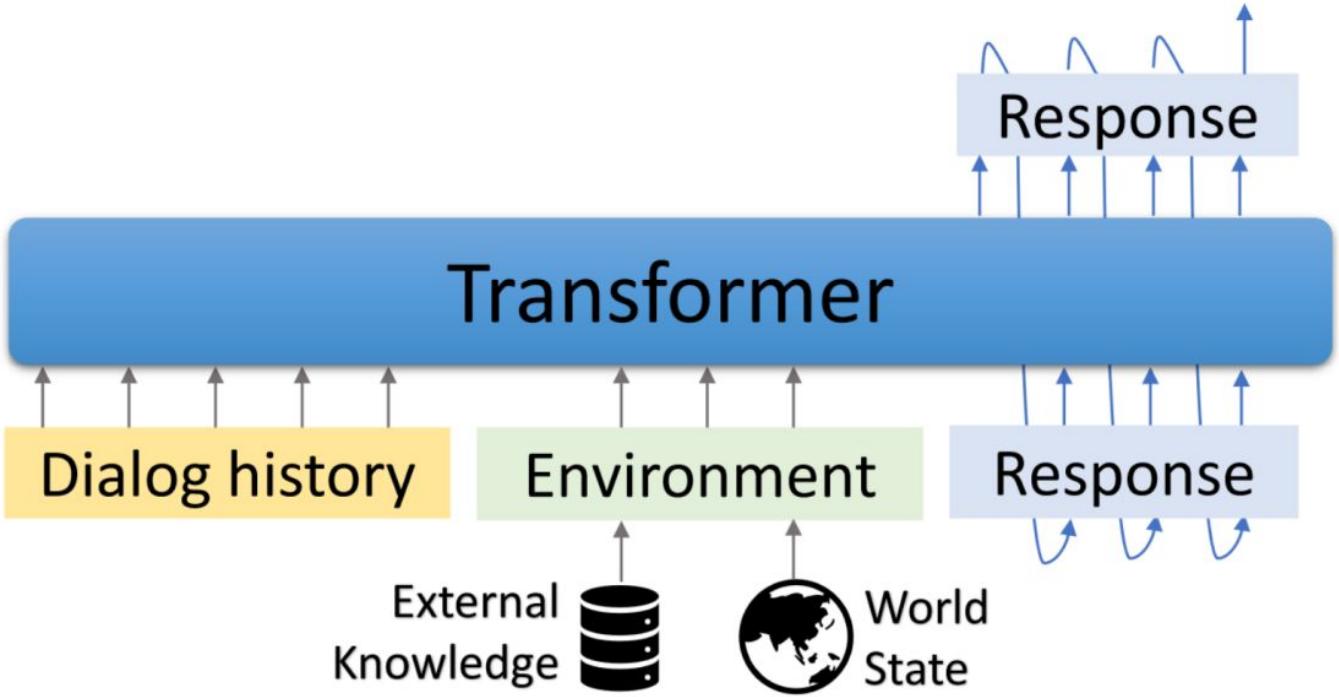
Classification Report:

	precision	recall	f1-score	support
chair	1.00	1.00	1.00	84
cobra	0.98	1.00	0.99	93
dog	1.00	1.00	1.00	84
tree	1.00	1.00	1.00	96
warrior	1.00	0.97	0.99	68
accuracy			1.00	425
macro avg	1.00	0.99	0.99	425
weighted avg	1.00	1.00	1.00	425

Model Evaluation



GODEL Model Architecture



Data Format for GODEL


```
{
    "INSTRUCTIONS": [
        "Instruction: given a dialog context and related knowledge, you need to response safely based on the knowledge.",
        "Instruction: given a dialog context, you need to response empathically.",
        "Instruction: given a dialog context and related knowledge, you need to answer the question based on the knowledge."
    ],
    "KNOWLEDGE": [
        "Regular exercise is essential for maintaining good health and fitness levels.",
        "Yoga is a holistic practice that combines physical postures, breath control, and meditation to promote overall well-being."
        "Joining a gym provides access to a wide range of fitness equipment and classes to cater to different exercise preferences."
        "Strength training exercises at the gym, such as weightlifting, help build muscle mass, increase metabolism, and improve bon
        "Cardiovascular exercises like running, swimming, or cycling improve heart health and boost endurance.",
        "High-intensity interval training (HIIT) workouts are effective for burning calories and improving overall fitness in a shor
        "Pilates is a low-impact exercise method that focuses on core strength, flexibility, and body awareness.",
        "CrossFit is a fitness program that combines elements of weightlifting, cardio, and bodyweight exercises to promote function
    ],
    "DIALOG": [
        "I am trying to do yoga? Tell me about it.",
        "yoga is a holistic practice that combines physical postures, breath control, and meditation to promote overall well-being."
    ]
}
```

API Response from GODEL

Request URL

```
http://localhost:5000/query/
```

Server response

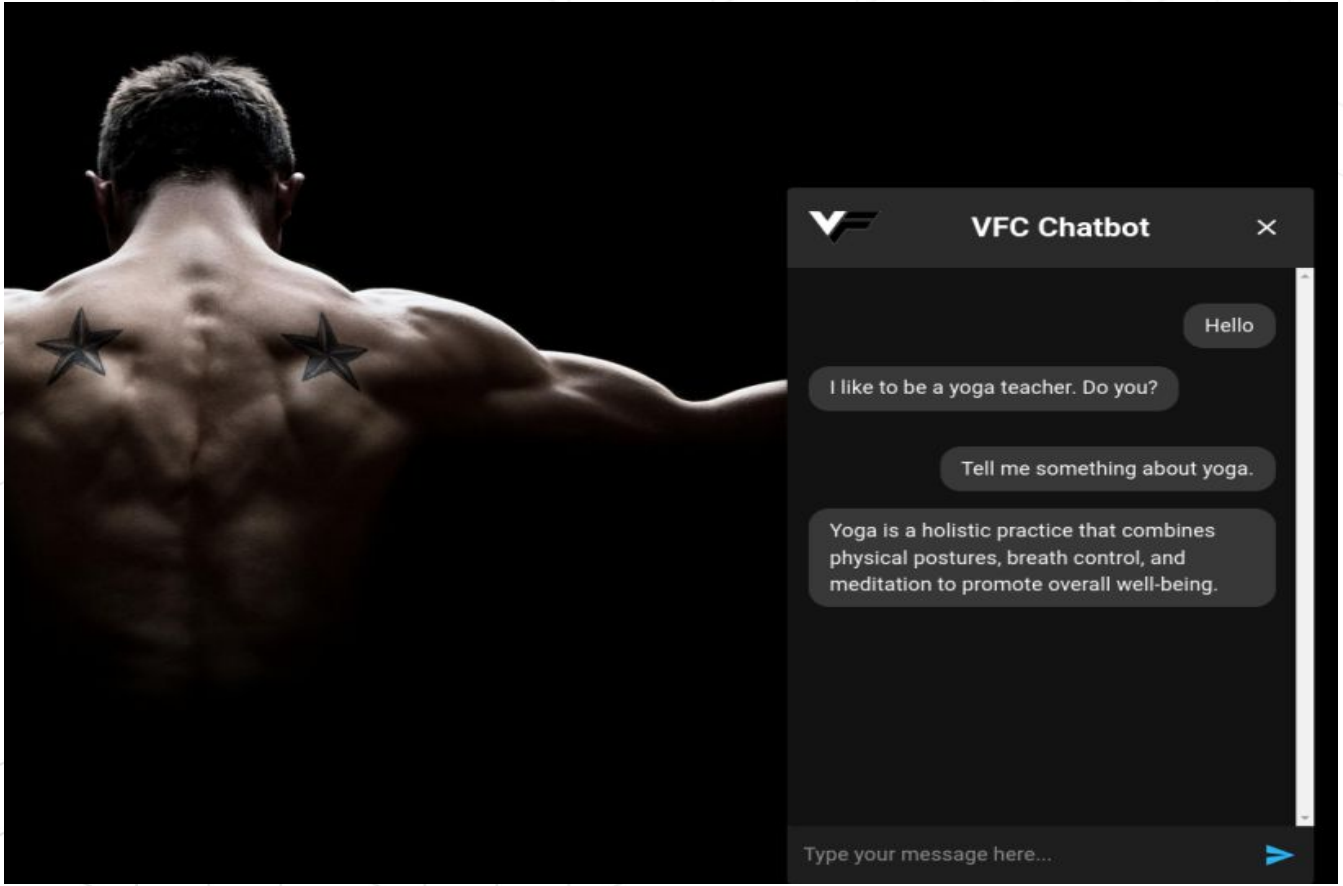
Code	Details
200	Response body

```
{
  "Question": "Tell me something about Yoga.",
  "Answer": "Yoga is a holistic practice that combines physical postures, breath control, and meditation to promote overall well-being."
}
```

Response headers

```
access-control-allow-origin: http://localhost:5000
content-length: 178
content-type: application/json
date: Thu,06 Jul 2023 15:20:36 GMT
server: uvicorn
vary: Origin
```

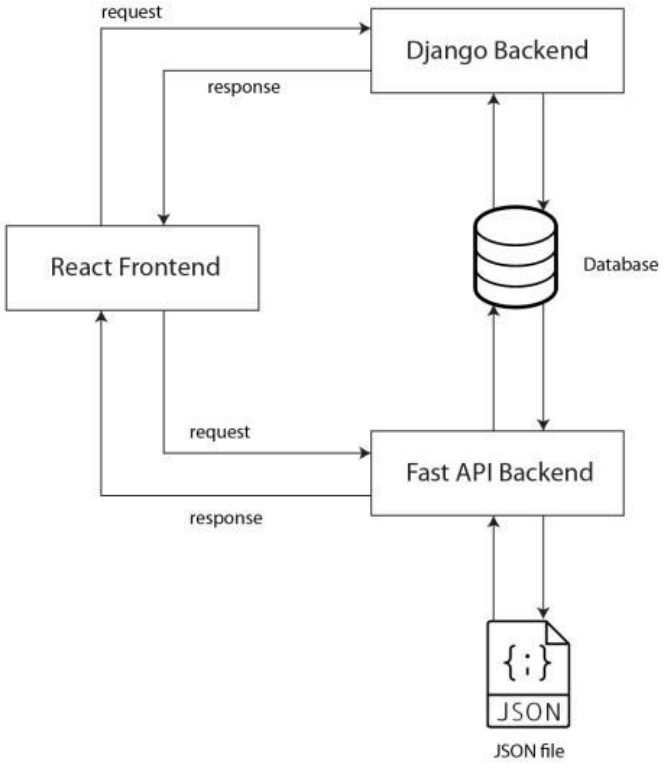
Chatbot demo



Performance Evaluation

1. Internal Criteria
2. Relative Criteria
3. External Criteria
4. F1 score
5. Precision
6. Recall
7. Confusion Matrix
8. Accuracy

Deployment Model of Virtual Fitness Coach



Expected Outcomes

- Personalized diet recommendations based on user data.
- Workout guidance.
- Performance monitoring and feedback.
- Provide user’s progress report.
- Personalized workout plan.
- Fitness Blog and Fitness Information Chatbot.
- Posture Tracking and Details.

Project Members and their Roles

- Asbin Khanal (Front End in React JS)
- Rohil Maharjan (Designer and Front end developer)
- Manita Kunwor (Backend developer in Django)
- Thomas Basyal (Project Manager, Machine Learning)
- Aayush Raj Regmi (System Analyst, Machine Learning)

Chore Assignment Structure

Virtual Fitness Coach

Front End

- Register
- Login
- Dashboard
- Fetch Diet
- Fetch Blog
- Blog Page
- Web View

Backend

- Sign Up API
- Login/ Logout
- User Profile API
- Post Blogs
- View Blogs
- Blog Details

Designer

- Dashboard Design
- System Design & Diagram
- Animation & visual effects
- React Component implementation
- CSS

Project Manager

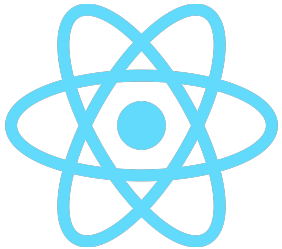
- Research & Development
- Team Lead
- Data survey
- Data Filtering, Processing
- Research
- Progress tracking

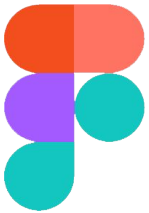
System Analyst

- Research & Development
- Pose Detection
- Object Oriented Analysis
- Backend Helper
- Machine Learning
- Research Design techniques and Implementation

Tools and Technologies

Front-End Tools





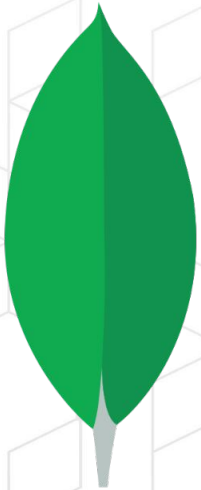
Figma

Tools and Technologies

Back-end Tools

Tools and Technologies

Database Tools



mongoDB®

Tools and Technologies

ML Techniques

- K-Means, KNN, CNN, mediapipe, Transformers, Hugging Face, Movenet, tensorflow, pytorch.
- GODEL for conversational AI (still doing research for alternatives)

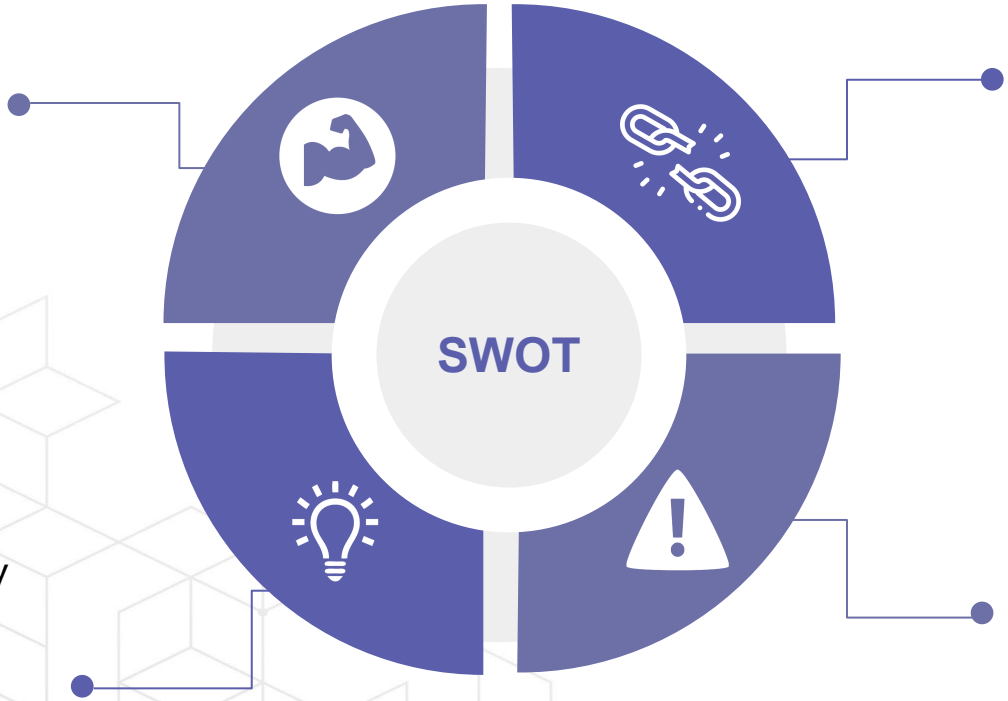
SWOT Analysis

Strengths

- Innovative and Highly Personalized Experience
- Convenient and Accessible 24 x 7
- Scalable Business Model
- Expertise and Guidance

Opportunities

- Growing Fitness Industry
- Rising Health Consciousness
- Strategic Partnerships
- International Expansion



Weakness

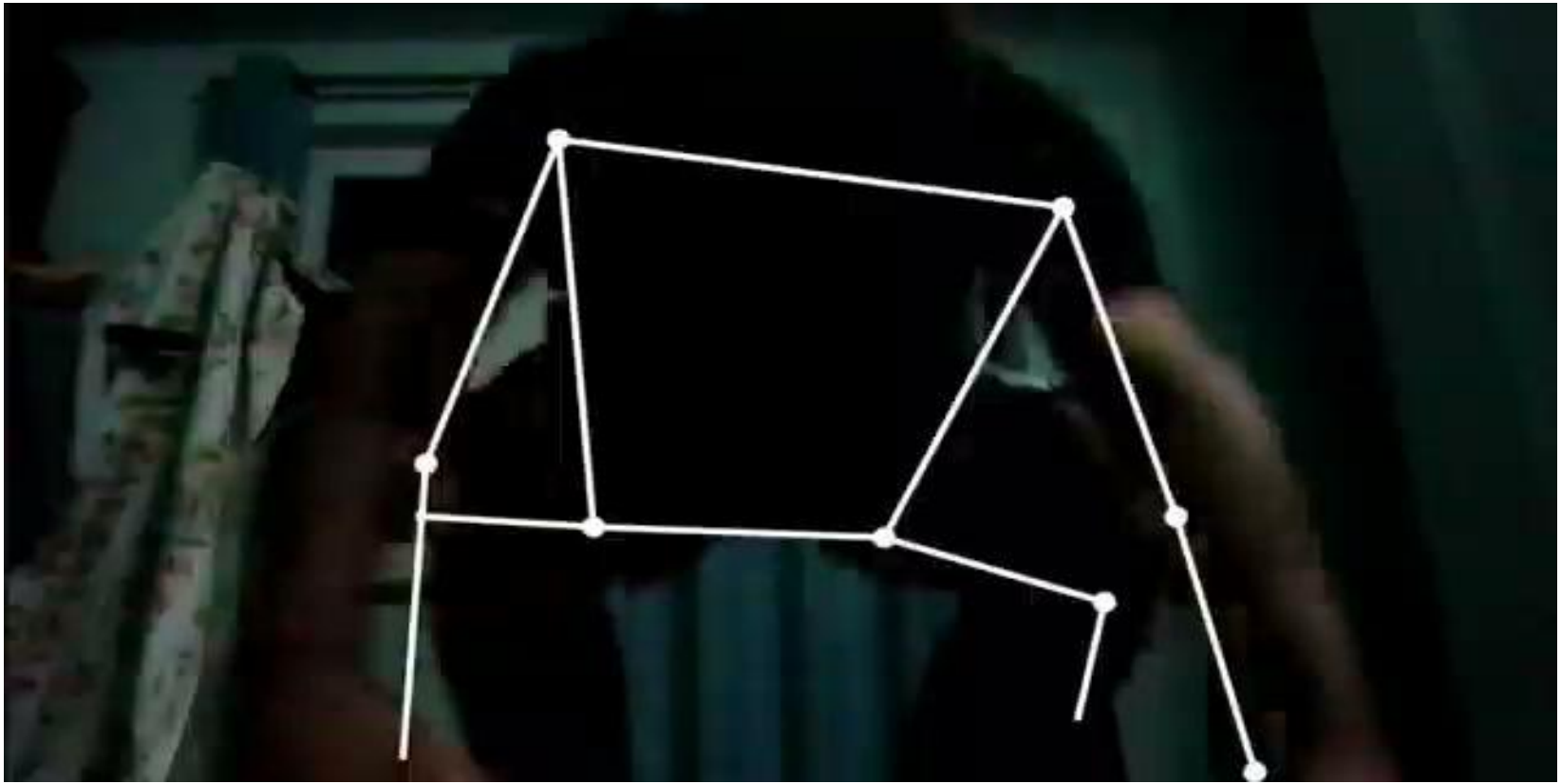
- Limited Brand Awareness
- Technological Dependencies

Threats

- Intense Competition
- Regulatory and Legal Considerations
- Technological Advancements
- Changing Consumer Preferences

वृक्षासन / Tree Pose

त्रिकोणासन / Triangle Pose



Thank You