RAG-LLM Healthcare Interaction System

Report submitted in partial fulfillment of the requirements
for the

B.Tech. in

Computer Science and Engineering Artificial Intelligence

By

Ojas (2021UCA1825)

Nikita Kanodia (2021UCA1803)

Under the supervision of

Prof. M.P.S. Bhatia

Computer Science and Engineering(CSE)

Netaji Subhas University of Technology, Delhi

Department of Computer Science and Engineering

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY
DELHI-110078

DECEMBER 2024

CERTIFICATE

This is to certify that the project titled **RAG-LLM Healthcare Interaction System** is a bonafide record of the work done by

Ojas (2021UCA1825)

Nikita Kanodia (2021UCA1803)

under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering** in **Information Technology** of the **Netaji Subhas Institute of Technology, University of Delhi, DELHI-110078**, during the year 2020-2021.

The original Research work was carried out by the team under my guidance and supervision in the academic year 2023-2024. This work has not been submitted for any other diploma or degree from any university. Based on the declaration made by the group, we recommend the project report for evaluation

DATE:

Prof Name

Designation

Division of Information Technology

Netaji Subhas Institute of Technology

University of Delhi

DECLARATION

This is to certify that the work which is being hereby presented by us in this project titled "**RAG-LLM Healthcare Interaction System**" in partial fulfilment of the award of the Bachelor of Engineering submitted at the Department of Information Technology, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, is a genuine account of our work carried out during the period from August 2020 to December 2021 under the guidance of ——- prof name———, Department of Information Technology, Netaji Subhas Institute of Technology, University of Delhi, New Delhi.

The matter embodied in the project report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

DATE:

Name 1
(Roll 1)

Name 2
(Roll 2)

ACKNOWLEDGEMENT

We would like to take this opportunity to acknowledge the support of all those without whom the completion of this project in fruition would not be possible.

——— write more here———

TABLE OF CONTENTS

List of Tables

List of Figures

type Introduction here

0.1 Section 1 of Intro

Cite like this [**?**]type Introduction here

0.2 Section 1 of Intro

Cite like this [**?**]

Chapter 1

Motivation

Chapter 2

Literature Review

—insert lit review—

Appendices

Appendix A

Code Attachments

A.1 Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

```
def get_parameters(data, chunk_size=410):
  #Store the activity label to add later
  activity = data['Activity']
    '''
    Define a dictionary of functions. Sets of readings will be
    aggregated as per these functions
    '''
  func_dict = {
    'min': np.min,
    'max': np.max,
    'diff': lambda x: np.max(x) - np.min(x),
    'std': np.std,
    'iqr': stats.iqr,
    'rms': lambda x: np.sqrt(np.mean(np.square(x))),
    'mad': lambda x: x.mad(),
    'mediad': mediad
  }
  aggregations = {
    'X': func_dict,
    'Y': func_dict,
    'Z': func_dict
  }
  data_groups = []
    '''
    Transform the dataset into rolling windows of 410 readings each
    and store them in a Pandas data group.
    '''
```

```
  for i in range(int(data.shape[0]/(chunk_size/2)) - 1):
    temp = data.iloc[int(i*(chunk_size/2)):int((i+2)*(chunk_size/2))]
    temp['k'] = i
    data_groups.append(temp)
  data_groups = pd.concat(data_groups).groupby('k', as_index=False)
    #Run the aggregations on all data groups
  stats_data = data_groups.agg(aggregations)
  stats_data.columns = [''.join(col).strip() for col in stats_data.
   columns.values]
  activity = activity.reset_index(drop=True)
    #Add activity label
  stats_data = pd.concat([stats_data, activity[:len(stats_data)]],
   axis=1)
  del stats_data['k']
  return stats_data
```