**Pseudocode** :

// Main Body/Program

Welcome to Cv Analyzer

// Start The System/Software

//Initialization of the User Interface

For Job Profile

While Loop ( True ) Do

Display The Main Menu

Get The User Choice() = User Choice

Switch Statement User Choice

Case : Create The job Profile

Create\_Job\_Profile()

Case : Upload Job Information/Description

Upload\_Description()

Case : Upload\_CV

Upload\_CV()

Case : View Ranked CVs

View\_Ranked\_CVs()

Case : Exit

Exit

End of the Switch Statement

End of the While Loop

End of the CV Analyzer Software

// Function for creating Job Profile

Function Create\_Job\_Your\_Profile()

Profile\_ID = Generate\_Unique\_Job\_ID()

Job\_Description = Get\_Employee\_Description()

Save\_Profile(Profile\_ID , Job\_Description)

Display\_Prompt\_Message : ”Profile Has Been Created Successfully”

End Of above Function

// Function for Uploading specific Description for specific Job

Function Upload\_Job\_Description()

Profile\_ID = Get\_User\_Choice(“ Data for accessing Specific Job Profile”)

Job\_Description\_File = Get\_User\_Input\_File(“eg : PDF, DOCX, TXT”)

Parsing\_Criteria = Parse\_Job\_Criteria( Job\_Description\_File)

Save\_Parsed\_Criteria = (Profile\_ID, Parsing\_Criteria)

Prompt\_Message = (“ Job Description Uploaded Successfully and has been parsed Successfully”)

//Function to upload CVs

Function Upload\_CVs()

Profile\_ID = Get\_User\_Choice(“ Data for accessing Specific Job Profile”)

CVs = Get\_User\_Input(“Upload Cvs eg : PDF, DOCX, TXT”)

For each CVs in CVs Do

Applicant\_ID = Assign\_Unique\_ID()

Extracted\_Text = Extraxt\_Important\_Text\_From\_Files(CVs)

Parsed\_Data/Text = Parsed\_CV(Extracted\_Text)

Save\_CV\_Data(Applicant\_ID, Extracted\_Text, Parsed\_Data/Text)

End of For Loop

Prompt Message = (“CVs have been Uploaded Successfully”)

//Function For Viewing Ranked CVs

Function View\_Ranked\_CVs()

Profile\_ID = Get User Input ( “ Select Job Profile “)

Ranked\_CVs = Ranked\_CVs(Profile\_ID)

Display\_Ranked\_CVs(Ranked\_CVs)

Return Parsed\_Criteria

End Function

//Function to extract Data from Various types of files

Function Extract\_Text(File)

If File Type is PDF Then

Return Extract\_Text\_Via\_PDF\_Box(File)

Else If File Type is DOCX Then

Return Extract\_Text\_Via\_DOCX\_Box(File)

Else If File Type is TXT Then

Return Simply\_Read\_Text\_File(File)

End If

End Function

//Function to Parse CVs

Function Parse\_CVs(Extracted\_Text)

Parsed\_Info = NLP\_Process(Extracted\_Text)

Return Parsed\_Info

End Function

//Function To Rank CVs Based on Job Criteria

Function Rank\_CVs(Profile\_ID)

Job\_Criteria = View\_Job\_Criteria(Profile\_ID)

CVs = View\_CVs(Profile\_ID)

Ranked\_CVs\_List [ ]

For each CV in CVs Do

Relevance\_Score = Calculate\_ Relevance\_Score(CVs, Job\_Criteria)

RankedList.ADD(CVs, Relevance\_Score)

End of For Loop

RankedList.Sort(Relevance\_Score Descending\_Order)

Return RankedList

End of Function

// Function To Save Data Securely

Function Save\_Data\_Securely(Data, File\_Name)

Encrypted\_Data = EncryptedDataWithAES256(Data)

Save\_To\_File(File\_Name, Encrypted\_Data)

End of Function

//Function For Displaying Ranked CVs

Function Display\_Ranked\_CVs(Ranked\_CVs)

For each CV IN Ranked\_CVs Do

Display CV.Summary and CV.Relevance\_Score

End of For Loop

End of Function

//Function For Encrypting Data

Function EncryptedDataWithAES256(Data)

//Do Implement AES-256 Encryption Logic

Return Encrypted\_Data

End of Function

// Function to Handle The User Input

Function Get\_User\_Input()

// Display and Return The User Input

End of Function

//Function To Display The Messages to The User

Function to display the Messages to the User

//Display The Messages to The User

End of The Function

Function To Assign The Unique IDs

Function Assign\_Unique\_IDs()

// Will Assign and Return the Unique IDs

End Of The Function

//Function To Save The Job Profile

Function Save\_Job\_Profile(Profile\_ID, Job\_Description)

//Will Store the Job Profile into The JSON file

End Of The Function

//Function To Save Parsing Criteria

Function Save\_Parsing\_Criteria(Profile\_ID, Parsing\_Criteria)

// Will Save The Parsing criteria into The

End Of The Function

//Function To Save CV Data

Function Save\_CV\_Data(Profile\_ID, Applicant\_ID, Parsed\_Data)

//Will Save The Data From The CVs Into The

End Of The Function

//Function To Load Job Criteria

Function Load\_Job\_Criteria(Profile\_ID)

//Will Load And Save The Job Criteria From The

End Of The Function

// Function To Load CVs

Function Load\_CVs(Profile\_ID)

// Will Load and Save CVs From The

End Of The Function

//Function To Calculate Relevance

Function Calculate\_Relevance\_(CV, Job\_Criteria)

// Will Implement The Logic To Calculate The Relevance On The Basis Of Matching Criteria

Return Relevance\_

End Of The Function

//Function To Perform NLP(Natural Language Processing)

Function NLP\_Process(Text)

//Implement NLP Processing Logic Using Stanford CoreNLP

Return Parsed\_Entities

End of The Function