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_Percentage = Calculate_ Relevance_Percentage(CVs, Job_Criteria)
        RankedList.ADD(CVs, Relevance_Percentage)
End of For Loop
        RankedList.Sort(Relevance_Percentage 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_Percentage
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 DataBase
```

End Of The Function

//Function To Save Parsing Criteria

Function Save_Parsing_Criteria(Profile_ID, Parsing_Criteria)

// Will Save The Parsing criteria into The DataBase

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 DataBase

End Of The Function

//Function To Load Job Criteria

Function Load_Job_Criteria(Profile_ID)

//Will Load And Save The Job Criteria From The DataBase

End Of The Function

// Function To Load CVs

Function Load_CVs(Profile_ID)

// Will Load and Save CVs From The DataBase

End Of The Function

```
//Function To Calculate Relevance Percentage
        Function Calculate_Relevance_Percentage(CV, Job_Criteria)
          // Will Implement The Logic To Calculate The Relevance
Percentage On The Basis Of Matching Criteria
        Return Relevance_Percentage
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
```