Projects as Final Assignment:
CS331

1. Professional Academia website
2. Android app development: Speech Translator
3. Clustering Visualization
4. GUI Based Resume Building Application
5. GUI based chat application using socket programming
6. Desktop Notification:
7. Submission Portal
8. Android app development: Split wise
9. Android app development: Jupiter

1. Professional Academia website

Project Title: Professional Academia website

Project Description: List of modules that you can include in the description of making the website:

- **Home Page:** The home page will introduce the visitors to the research lab and its activities. It can include a brief overview of the lab, its mission, and objectives.
- **About Us:** This section will provide detailed information about the research lab, including its research activities, and achievements.
- **Research:** This module will showcase the research work of the lab, including its current projects and published papers. It can also highlight the research facilities and resources available at the lab.
- **Events:** This section will highlight the upcoming events related to the lab, such as seminars, conferences, workshops, and guest lectures.
- **Resources:** This section will provide resources related to research domain and language, such as links to online databases, tools, and software.
- **Contact Us:** This module will contain contact information for the research lab, including its address, phone number, email address, and social media handles.

- **Publications**: This module will include a list of publications by the lab members, including papers, books, and chapters.
- **People:** This section will contain profiles of the researchers, including their academic and research backgrounds.
- **Collaborations:** This module will showcase the collaborations of the lab with other institutions, universities, and research organizations.

Note: These are some of the modules that you can include in the description of making the website for the Professional academia website. However, you can also customize the modules according to your requirements and preferences.

2. Android app development: Speech Translator

Project Title: Speech Translator

Project Description:

This project aims as an extension of existing algorithm that we will provide you.

- **Necessary Implementation:**
  1. **Speech Recognition:** An algorithm code will be provided to you, you task is to extend and add some functionalities to it.
  2. **Text-to-Speech Synthesis:** Implement a text-to-speech synthesis module that can convert the translated text from the machine translation module into speech output in the target language.
  3. **Real-time Processing:** Ensure that the system is capable of processing speech input and producing translated speech output in real-time.
- **Extensions:**
  1. **User Interface:** Develop a user-friendly interface that allows users to easily input their speech in the source language, select the target language, and receive the translated speech output
  2. **Language Support:** Implement support for multiple source and target languages

Note: Overall, the Speech Translator project requires the integration of speech recognition, machine translation, and text-to-speech synthesis technologies, along with a user-friendly interface and rigorous testing to create an effective and efficient system for real-time speech translation.

3. Clustering Visualization

Project Title: Affinity Propagation Visualization Tool

Project Description:

The goal of this project is to develop a **GUI based application** that will help users understand how the Affinity Propagation algorithm works by visualizing each iteration. The Affinity Propagation algorithm is a clustering algorithm that works by sending messages between data points to determine which points should be considered as exemplars, and which points should be assigned to those exemplars.

The application will take input data in the form of a matrix or dataset, and will allow the user to set the parameters of the algorithm such as damping factor, preference value, and maximum number of iterations. The application will then display a visualization of the clustering process in real-time, allowing the user to see how the algorithm is working at each iteration.

The application will display a scatter plot of the data points and will highlight the exemplars in different colors. The user will be able to see the messages being sent between points, and how these messages are influencing the clustering process. The user will also be able to see the decision-making process of the algorithm, such as which points are being considered as exemplars, and which points are being assigned to those exemplars.

Possible Features:

- Input data in the form of a matrix or a dataset
- Setting of algorithm parameters such as damping factor, preference value, and maximum number of iterations
- Real-time visualization of the clustering process

- Scatter plot display of data points with different colors for exemplars
- Display of messages being sent between points
- Display of decision-making process of the algorithm
- Save and load visualization files
- Help and documentation

Deliverables:

- A GUI based application that visualizes each iteration of the Affinity Propagation algorithm
- Documentation of the project, including user manual and technical documentation
- Sample datasets and visualizations for testing and evaluation

Possible Extensions:

- Adding support for other clustering algorithms such as K-Means or DBSCAN
- Adding support for other input formats such as CSV or Excel files
- Adding support for other visualization techniques such as 3D scatter plots or heatmaps
- Adding support for different color schemes and themes

4. GUI Based Resume Building Application

Note: *Any programming language that the student finds helpful, can be used.*

Build a GUI based Resume building Application. The Application must have a Page-based view, details of which will be mentioned below.

The first Page[Page - 0] must be a blank one, containing a big button titled: **BUILD YOUR RESUME.** On clicking which next page should open. You may customize this page more.

GUI Interface should have the following fields:-

1. Personal Details: This field will have 3 sub-fields. All 3 fields: Name, Upload Photo and Contact Details must be on same page.[Page - 1]

    a. Name*: This field will give the Name of individual to the final draft.

    b. Upload Picture: This field will upload the picture to the final draft. If no picture is chosen just upload a default random plain picture.

    c. Contact details:
        i. Mobile number*. (Add error check of 10 digit and numeric data).
        ii. E-mail*. (Add @ error check)
        iii. Current Address*.

On entering the details in each of the fields we will get corresponding detail in final draft.

There should be a button **Save & Next** to add the details.

Once Save & Next is clicked, a new page shall open that ask for next details.

2. Professional Details: All the sub-fields must be on different page, when one page is clicked, it should go to the next. This field will have following sub-fields:

d. Professional Summary*: This field will describe the person. (Add error checking that it should be atleast 10 words long). After Professional Summary is written, a click on **Save & Next** button should be made to go further. It must open new page closing the current one.[Page - 2]

e. Skills: This field will describe the person's skills and insert to the final draft. This field must have a **ADD More Skill** button which will add more and more data. (Add an error check here that will not allow a size of text more than 30 for each skill). After all the Skills are entered, a click on **Save & Next** button should be made to go further. It must open new page closing the current one.[Page - 3].

f. Experience: Before adding experience, there should be a **Skip** button, since the students may not have any experience. This

field will describe the person's experience and insert to the final draft. It should have following sub-fields:[Page - 4]

i. Job title*.
ii. Employer*.
iii. Start date*.
iv. End date*.

This field must have a **ADD More** button which will add more and more data. Once relevant data is inserted, a click on **Save & Next button** must be made. It must open new page closing the current one.

3. Education Details: This field will describe the person's educational background and insert to the final draft. This field must have a **ADD More** button which will add more and more data. It should have following sub-fields: [Page - 5] (Atleast one educational detail is necessary).

g. School Name*.
h. Degree*.
i. Field of study*.
j. Start date*.
k. Graduation date*.

On clicking Save & Next button, a new page must open.

4. Achievements & Certifications: Add a **Skip** button at top of both pages. Both Achievements and Certificates must be on different page. To enter details:[Page - 6, 7]

l. Achievements*: Add a text field, beside that also add **ADD More** button.

On clicking **Save & Next Button,** open new page.

Certificates: Also add **ADD More** button.

m. Certificate Name field*.
n. Certified By field*.

On clicking **Save & Next Button,** open new page.

[Page - 8]Atlast give an option to save the Resume as a PDF file. **A SAVE RESUME** button.

The * represents the mandatory field without which the Save & Next Button should NOT work. All the date field must open a calender view. No Manual entry of date allowed. Apply other necessary error checks on fields(if applicable).

Note: *You may add other things as per your creative mind. But make sure that these basic things are present.*

The final Resume(Saved one) must look like this. Although you are free to use your own creativity, make sure that the basic things mentioned above must be present. Only addition to above points are allowed, NO field mentioned above must be deleted.

HARSH BIJWE

+91-8234074223
harshbjw26@gmail.com
Room A-310 Barak Hostel IITG Campus,
Guwahati, Assam 781039 India

PROFESSIONAL SUMMARY

I'm Harsh Bijwe. I am currently pursuing my M. Tech in Computer Science from IIT Guwahati, having done my B.Tech from CCET Bhilai. After scoring a decent rank in GATE 2021, I got the opportunity to study in one of the prestigious institutes of our nation. I am interested in Software development and have worked on many interesting projects like User Preference Aware Fake News Detection and Parking Management System using Automatic Number Plate Detection. For more details, visit my profile: https://github.com/Harsh26

SKILLS

- Technical Skills
- Programming: C/C++, Java, Python, Prolog
- Tools/Frameworks used in Projects:Pytorch, OpenCV
- Operating Systems: Windows, Linux

EXPERIENCE

Student , **IIT Guwahati**, Jul 2021 - Current, Guwahati, Assam

- I am currently pursuing my studies in the Computer Science and Engineering Dept. of IIT Guwahati. I also work as Teaching Assistant here.

EDUCATION

M.Tech
Indian Institute of Technology - Guwahati
GPA: 7.77

B.Tech Sep 2020
Christian College of Engineering and Technology - Bhilai
GPA: 9.07

High School Diploma, Physics, Chemistry and Math Apr 2016
DAV Public School - Bhilai, CG
Percent: 90.8%

5. GUI based chat application using socket programming

Project Name: GUI-Based Chat Application

Project Description: In this project, you will create a GUI-based chat application. The chat application should allow multiple users to connect and communicate with each other in real-time. The application should have a user-friendly interface that allows users to join or create chat rooms and send messages to each other.

Project Requirements:

User Interface Design: Design an intuitive user interface that allows users to easily join or create chat rooms and send messages to each other.

User Authentication: Implement a user authentication system that allows users to register and log in to the chat application. The user authentication system should also allow users to reset their passwords.

Chat Rooms: Implement a chat room system that allows users to create or join chat rooms. Each chat room should have a unique name and should display a list of all the users currently in the room.

Real-time Messaging: Implement a messaging system that allows users to send and receive messages in real-time.

Chat History: Implement a chat history system that allows users to view the conversation history of a chat room.

Error Handling: Implement error handling to handle unexpected situations such as network errors and user input errors.

Data Persistence: Implement a data persistence system that allows users to save their chat history and login credentials.

Deliverables:

Source Code: Submit the source code of the chat application.

User Manual: Provide a user manual that explains how to use the chat application.

Project Report: Write a project report that explains the design and implementation of the chat application. The report should also include screenshots of the chat application and a description of the testing process.

Additional Features (Optional):

Emojis: Implement an emoji system that allows users to send emojis in their messages.

Notifications: Implement a notification system that alerts users when they receive a new message or when they are mentioned in a chat room.

File Sharing: Implement a file-sharing system that allows users to share files with each other.

6. Desktop Notification

Project Title: Desktop Notification

Description:

The Desktop Notification project aims to develop a system that can send notifications or alerts to a user's desktop or notification center on their computer. These notifications can be used for various purposes, such as reminders, updates, events, or important information. The system will display the notifications, providing a convenient way for users to receive important notifications without having to constantly check their emails or other communication channels.

- **Necessary Implementation:**

1. Notification Triggering: Implement a module that can trigger notifications based on predefined events, actions, or conditions. This could include setting up notifications for calendar events, incoming messages, system alerts, or custom events defined by the user.

2. Notification Display: Develop a visually appealing and non-intrusive notification display that appears on the user's desktop or notification center. The notification should include relevant information, such as the title, content, and source of the notification, and be displayed in a way that is easy to read and dismiss.

3. User Interaction: Implement user interaction features, such as buttons or links within notifications, that allow users to take actions directly from the notifications, such as dismissing or snoozing

notifications, opening related applications or websites, or responding to messages.

4. Notification Management: Develop a notification management module that allows users to view, dismiss, and organize notifications in a convenient way, such as grouping notifications by source, category, or priority, and providing options for clearing or archiving notifications.

5. Error Handling: Implement error handling mechanisms to handle situations such as failed notifications, network issues, or system errors, and provide appropriate feedback to the user.

- **Possible Extensions:**

1. Customization Options: Provide options for users to customize the appearance and behavior of notifications, such as choosing different notification styles, setting priority levels, or defining notification preferences based on specific criteria.

2. Platform Compatibility: Ensure that the desktop notification system is compatible with various desktop operating systems, such as Windows, macOS, and Linux, and can be easily integrated into different desktop environments or notification centers.

7. Submission Portal

The aim of this project is to create an assignment submission portal for a particular course. The details of the project are given below:

User Authentication: The portal should require students to log in using their university or college credentials to access the portal. This ensures that only registered students can submit assignments.

Assignment Creation: Faculty members should be able to create assignments and upload relevant documents or files, including assignment instructions, rubrics, and examples.

Submission Form: Students should be able to submit their assignments through a submission form provided on the portal. This form should have fields for students to enter their name, student ID, course name, assignment title, and the file they want to submit.

Submission Deadline: The portal should have a feature for setting a submission deadline for each assignment. This will allow students to submit their assignments before the deadline and prevent them from submitting assignments after the deadline.

Grading and Feedback: After the submission deadline has passed, faculty members should be able to access submitted assignments, grade them, and provide feedback to students. Students should also be able to view their grades and feedback on the portal.

Notifications: The portal should have a notification feature that alerts students and faculty members of upcoming assignment deadlines, grades, and feedback.

Note: You can also customize and add the modules according to your requirements and preferences.

8. Android app development: Split wise

Overview:

Splitwise is an expense-sharing application that simplifies the process of splitting expenses between friends, family, and roommates. In this assignment, you will design and develop a similar app that can keep track of expenses, split them between different groups and individuals, and allow users to settle debts.

Requirements:

Your app should have the following modules:

User Registration and Login: Users should be able to create accounts and log in to access the app's features.

Dashboard: The dashboard should display a summary of the user's activity, including the total amount of expenses, debts, and balances.

Expense Management: Users should be able to add expenses and categorize them based on their purpose, date, and amount.

Group Management: Users should be able to create groups and invite other users to join them. Each group should have a separate dashboard that displays its members, expenses, and debts.

Expense Splitting: When adding an expense, users should be able to split it between different group members. The app should support multiple splitting methods, such as equally, by percentage, or manually.

Debt Management: The app should keep track of each user's debts and credits and allow them to settle them through various payment methods.

Notification and Reminder: The app should send notifications to users when they owe money to someone or when someone owes them money. Users should also be able to set reminders for upcoming payments.

Reports and Analytics: Users should be able to generate reports and analyze their spending habits based on different criteria, such as date range, category, and group.

Technical Requirements:

Your app should be developed using a modern programming language and should be compatible with both Android and iOS platforms. You can use any database management system for data storage and should follow the standard design patterns and practices for building scalable and maintainable applications.

Deliverables:

You should submit the following deliverables as part of your assignment:

Source Code: The app's source code, including all the necessary files and dependencies.

User Manual: A user manual that explains how to use the app's features and functionalities.

Demo Video: A demo video that showcases the app's key features and functionalities.

Note: The above requirements are just guidelines, and you are free to add additional features and functionalities to the app based on your creativity and innovation.

9: Android app development: Jupiter

Overview: Jupiter is a personal finance management application that simplifies the process of tracking and categorizing expenses. In this assignment, you will design and develop a similar app that can take an excel file as input and categorize expenses based on their purpose.

Requirements: Your app should have the following modules:

1. **User Registration and Login:** Users should be able to create accounts and log in to access the app's features.
2. **File Import:** Users should be able to upload an excel file containing their expenses. The app should parse the file and extract relevant information such as date, amount, and description.
3. **Expense Categorization:** The app should categorize each expense based on its purpose. You can use pre-defined categories such as food, transportation, rent, and utilities or allow users to create their own categories.
4. **Dashboard:** The dashboard should display a summary of the user's activity, including the total amount of expenses, debts, and balances. The dashboard should also display graphs and charts that show the user's spending habits and patterns.
5. **Reports and Analytics:** Users should be able to generate reports and analyze their spending habits based on different criteria, such as date range, category, and expense type.
6. **Budgeting:** The app should allow users to set budgets for different categories and send notifications when they exceed their budget limits.
7. **Notification and Reminder:** The app should send notifications to users when they exceed their budget limits or when they have upcoming payments.

8. **Settings:** The app should allow users to customize their preferences and settings, such as currency, date format, and language.

Technical Requirements: Your app should be developed using a modern programming language and should be compatible with both Android and iOS platforms. The app should support excel file import and parsing and should use machine learning or artificial intelligence algorithms for expense categorization. You can use any database management system for data storage and should follow the standard design patterns and practices for building scalable and maintainable applications.

Deliverables: You should submit the following deliverables as part of your assignment:

1. **Source Code:** The app's source code, including all the necessary files and dependencies.
2. **User Manual:** A user manual that explains how to use the app's features and functionalities.
3. **Test Plan:** A test plan that includes the test cases, test scenarios, and expected outcomes.
4. **Demo Video:** A demo video that showcases the app's key features and functionalities.

Note: The above requirements are just guidelines, and you are free to add additional features and functionalities to the app based on your creativity and innovation.