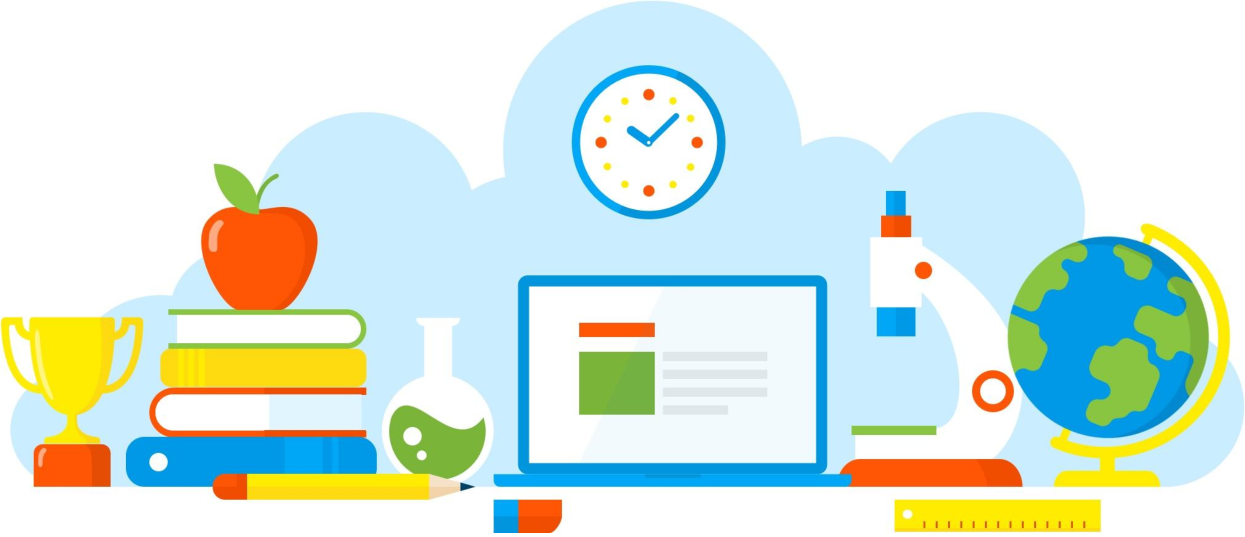

# SOLVE FOR IITK

*PROJECT DESCRIPTION*

TEAM STARSAILOR

## About the Problem Statement

Identify and formalize one of any of the below

- A challenge or problem faced by any part of the campus community.
- A system that does not yet exist for the campus, but would be of great benefit if it did.
- An inefficiency in the workflow or pipeline in a system implemented anywhere in the campus.

Define and ideate a mechanism to resolve/create/ameliorate the entity identified above through code.

Create a working proof of concept that your innovation works and can integrate itself into campus life.

## Problem Identified

Our team identified that the task of entering and exiting from the campus gate is a very tedious task. Often people forget updating the records, leading to consequences later. Many times, we are lost among the pages for finding our entry, resulting in wastage of time as well as patience. We wish to automate this process to fasten the process, helping the people of IIT Kanpur.

## Libraries used

- Pyttx3
- Face_recognition
- Cv2
- Numpy
- Io
- Speech_recognition

## Classes made

- Employee
- Visitor
- Student

## Brief description of the working code

1. The script imports the necessary modules, including `pyttsx3` for text-to-speech synthesis, `face_recognition` for face detection and recognition, `cv2` for computer vision operations, `numpy` for numerical operations, `io` for input/output operations, and `speech_recognition` for speech recognition.

2. The code defines a function called `facedetect` which performs face detection and recognition. It captures a frame from the webcam, encodes it as a JPEG image, and uses the `face_recognition` module to compare the captured face with a set of known faces. It returns the name of the recognized person or "Unknown" if the face is not recognized.

3. The code defines another function called `textfromspeech` which uses the `speech_recognition` module to convert speech input from the microphone into text. It adjusts for ambient noise, listens for input, and then uses Google's speech recognition service to transcribe the audio into text.

4. The code creates instances of different classes such as `Student`, `Employee`, and `Visitor`, and sets their attributes accordingly.

5. The `object_type` function is defined to determine the type of user (student, employee, or visitor) based on the input name. It iterates through the lists of students, employees, and visitors and checks if the input name matches any of their names. If a match is found, it sets the `user_type` variable accordingly and assigns the index of the matched object to the `user_index` variable.

6. The `main` function is defined to handle the main logic of the program. It starts by calling the `facedetect` function to recognize the user based on their face. It

then calls the `object_type` function to determine the user type and index.

7. Based on the user type and recognition status, the program performs different actions. For example:

- If the recognized user is a student and their `inside_status` attribute is `True`, it asks for their desired destination, updates the student's `visit_place` attribute, sets the `inside_status` to `False`, and provides an audio message to enjoy the visit.

- If the recognized user is a student and their `inside_status` attribute is `False`, it welcomes the student back to the campus.

- If the recognized user is an employee and their `inside_status` attribute is `True`, it provides an audio message for a happy journey and sets the `inside_status` to `False`.

- If the recognized user is an employee and their `inside_status` attribute is `False`, it welcomes the employee back to the campus.

- If the recognized user is a visitor, it provides an audio message for successful recognition.

8. The `main` function is called at the end of the script to start the program.

The code demonstrates a basic access control system where students, employees, and visitors are recognized based on their faces and provided with different actions and messages based on their status and type.

## Problems faced currently

- The accuracy of the model is not up to the level we were expecting. We were working on improving the same but could not do that due to time constraints.
- As of now, the prediction time is more than we expected. With a larger database, this time is expected to increase. We wish to work upon the same in the near future.

## Future Prospects

- We wish to refine the system for  the entry and exit of visitors.We will modify our system for on the spot registration of new visitors which will verify them with a government issued id card.
- We will also use a remote database such as Postgre SQL so that multiple devices can be installed to manage the heavy ingoing and outgoing of people.
- We will consider adding a graphical user interface (GUI) to provide a more intuitive and user-friendly interaction with the system. The GUI can display information, capture faces, and provide feedback to the user.
- We will consider using more efficient algorithms or techniques, such as face clustering, to improve the performance of face recognition.
- Address security concerns related to face recognition, such as data privacy, authentication mechanisms, and potential vulnerabilities. Implement secure practices to protect user information and prevent unauthorized access.

## Team Members

1. DHRUV MITTAL(220363)
2. HARDIK JINDAL(220420)
3. NIKHIL GUPTA(220708)
4. RAGHAV MANGLIK(220854)