## COMPANY PROFILE

**Company name: EZ Technologies Locations**: Hyderabad.

**Introduction:**

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

**Mission:**

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.Below is a brief insight into them.

**Services:**

**College Trainings:**

• Tailored training programs designed to enhance the employability of students.

• Industry-aligned curriculum covering technical and soft skills.

• Placement assistance and career guidance.

**Development Projects:**

• End-to-end development services, from ideation to execution.

• Expertise in diverse technologies and frameworks.

• Custom solutions to meet specific business needs.

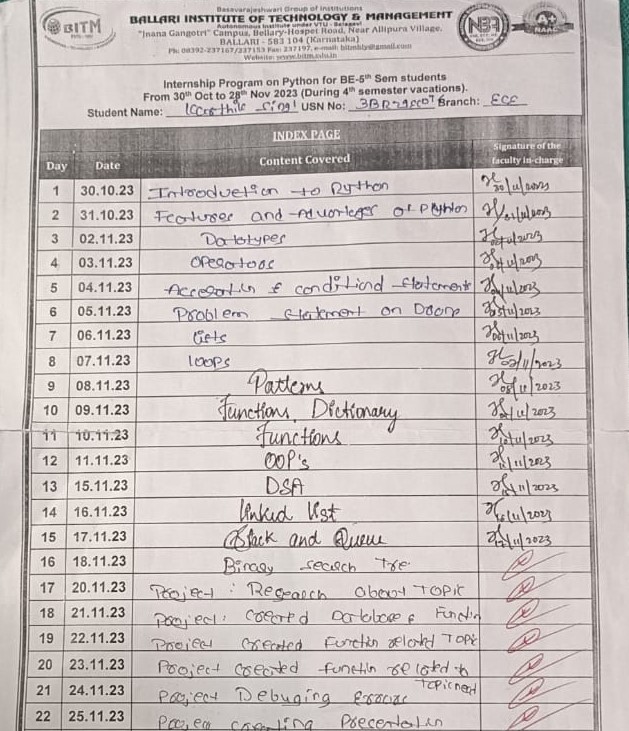
**Locations**: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

**ABSTRACT**

The abstract for a dynamic user onboarding system in Python could highlight the development of a flexible and interactive process for welcoming and guiding users into a software application. This system adapts to users' preferences, providing a personalized experience that enhances user engagement and accelerates the onboarding journey. The abstract may touch on the utilization of Python's versatility in creating a responsive onboarding flow, incorporating user feedback, and dynamically adjusting instructional content based on user interactions. Overall, the focus is on optimizing the onboarding experience through a dynamic and user-centric approach using Python.

**DAY TO DAY ACTIVITY:**



**INTRODUCTION:**

Dynamic user onboarding is a personalized and adaptive process designed to welcome and guide users as they familiarize themselves with a new product, service, or platform. Unlike static onboarding approaches, dynamic onboarding tailors the user experience based on individual characteristics, preferences, and interactions. The goal is to optimize user engagement, reduce friction, and enhance the overall user journey. It delves into the key components of dynamic onboarding, such as personalized content, interactive guidance, and continuous user feedback, highlighting their role in creating a seamless and evolving user journey. The abstract underscores the dynamic nature of user onboarding as a responsive mechanism that fosters a deeper connection between users and the product, ultimately contributing to sustained user engagement and loyalty. In this case study, we explore the implementation and impact of dynamic user onboarding in a mobile application designed for task management. The goal was to enhance user engagement and retention by tailoring the onboarding experience to individual user behaviors and preferences

## MODULE DESCRIPTION

This Python script defines a FeedbackSystem class with methods to collect, update, delete, and print feedback for different roles in an onboarding system. The onboarding system is designed for various roles, and the feedback is collected for different steps in the onboarding process.

Here's a brief description of the key components:

**1.’roles’ Dictionary:**

* Map’s role indices to role names (e.g., 1 is 'admin', 2 is 'employee', etc.).

**2.’FeedbackSystem’ Class:**

* **Attributes:**

l)Onboarding\_system: A nested dictionary defining the onboarding steps for each role.

II)Feedback\_data: A dictionary to store feedback for each role.

* **Methods:**

\_init\_: Initializes an instance of the FeedbackSystem class.

design\_onboarding\_flow: Prints the onboarding steps for a given role category.

collect\_feedback\_step: Collects feedback for a specific role.

update\_feedback: Updates existing feedback for a specific role.

delete\_feedback: Deletes feedback for a specific role.

print\_feedback: Prints a summary of all collected feedback.

print\_onboarding\_system\_key: Prints the category name for a given category index.

**3.main Function:**

* Creates an instance of the FeedbackSystem class and enters a loop to display a menu of options for the user.
* The user can choose to collect, update, delete, or print feedback, view onboarding steps, or exit the program.

**4.Menu Logic:**

* The user's input is processed using the match statement (introduced in Python 3.10).
* The program validates the user's input and performs the corresponding action based on the selected menu option.

**5.Usage:**

* The script is executed when the module is run, and the main function is called.

## FLOW CHART:

Onboarding\_steps

stop

exit

Invalid choice

Collect\_feedback

Print\_feedback

Delete\_feedback

Update\_feedback

Whileloop(choice)

1.collect feedback

2.update feedback

3.delete feedback

4.print feedback

5.To see onboarding steps

6.exit

start

ALOGRTHIM:

1. Initialize Data Structures:

* + Define roles and onboarding steps in a dictionary.
  + Create a class FeedbackSystem with an \_\_init\_\_ method to initialize feedback data.

2. Define Methods:

* + design\_onboarding\_flow(category): Display onboarding steps for a given category.
  + collect\_feedback\_step(role): Collect feedback for a specific role and store it in the feedback data.
  + update\_feedback(role): Update existing feedback for a role.
  + delete\_feedback(role): Delete feedback for a role if it exists.
  + print\_feedback(): Display a summary of all collected feedback.
  + print\_onboarding\_system\_key(category\_index): Print the category name based on the index.

3. Main Loop:

* + Create an instance of FeedbackSystem.
  + Use a while loop to repeatedly display a menu and accept user input.
  + Based on user input, perform actions such as collecting feedback, updating feedback, deleting feedback, printing feedback, displaying onboarding steps, or exiting the program.

4. Handle User Input:

* + Use a match statement (available in Python 3.10 and later) to handle different choices.
  + For choices 1-4, prompt the user for the onboarding category index, validate it, and then perform the corresponding action.
  + For choice 5, prompt the user for the category to see onboarding steps and call design\_onboarding\_flow.
  + For an invalid choice, display an error message.

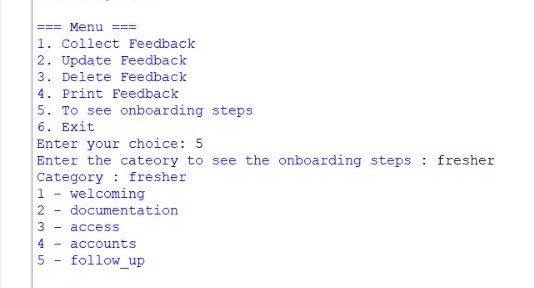
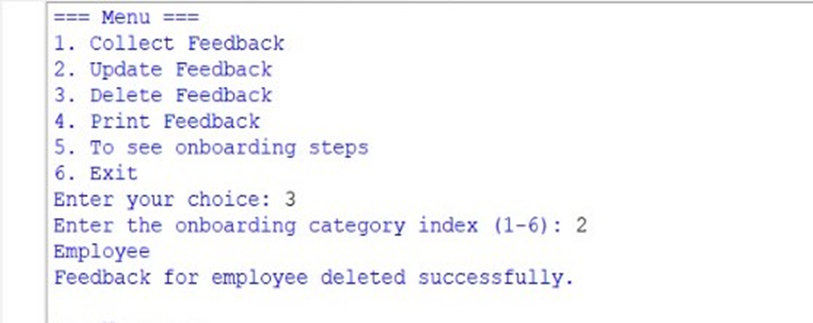
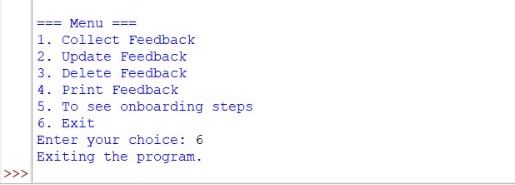
5. Run the Program:

* + In the \_\_main\_\_ block, create an instance of FeedbackSystem and call the main function.

6. Exit Program:

* + If the user chooses to exit (choice 6), break out of the loop and print an exit message.

## OUTPUT:



**CONCLUSION:**

In conclusion, dynamic user onboarding is about making the onboarding journey personal, adaptable, and efficient. By tailoring the experience to each user's needs, it enhances engagement, learning, and overall user satisfaction, ultimately contributing to higher retention rates and a more successful user onboarding process.

Dynamic user onboarding is a process that adapts to each user's needs and preferences to create a smooth and personalized experience.

**FUTURE SCOPE:**

The future of dynamic user onboarding involves leveraging advanced technologies like AI and machine learning to create highly personalized and adaptive user experiences. This entails tailoring onboarding processes in real-time based on user behavior, preferences, and contextual data. Expect continued innovation in streamlining and automating onboarding, ensuring a smoother and more engaging journey for users. As technology evolves, the integration of novel approaches, such as biometrics or blockchain, may contribute to enhancing security and user trust in the onboarding process.

**REFERENCES**

1. <https://.com/c/a2563e05-778f-47fb-9fdf-d1f779403be2>
2. <https://support.dynamicyield.com/hc/en-us/articles/360021867314-User-Data-Onboarding-by-CSV>