Info a!dTech

Connect With US:

Company Goal

The goal of Info aidTech internship program is to equip students with the skills and experience needed to succeed in the field of IT. Through our comprehensive internship program, we aim to provide students with a diverse range of IT experiences and opportunities, allowing them to explore various IT functions and technologies.

Our goal is to prepare our interns for successful careers in the IT industry by providing them with practical skills, real-world experience, and valuable industry insights. We are committed to helping our interns grow both personally and professionally, and to building long-lasting relationships that extend beyond the duration of the internship.

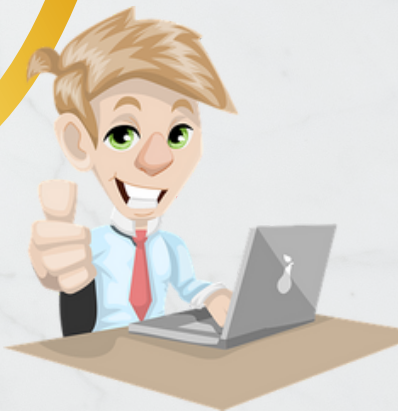
Instructions

- You have to update your LinkedIn profile & share your documents (Selection Letter, Internship Completion Certificate) and tag Info aidTech LinkedIn page and use relevant hashtags.(#infoaidtech, #infoaid)
- Maintain a separate GitHub repository.(Ex: aidTec_task name)
- You have to share GitHub repository link and video of completed tasks in LinkedIn and tag Info aidTech LinkedIn page with relevant hashtags.
- Share a proper video of Completed task on LinkedIn also tag us.
- For getting internship completion certificate you will need to complete minimum two task as per your respective domain.
- For getting LOR(letter of recommendation) you will need to complete all the task as per your respective domain and Peer Evaluation (Evaluate at least 3 tasks posted by fellow interns on LinkedIn).
- Make sure you follow us on LinkedIn, Instagram, Facebook and joined Telegram group.(Link attached in first page)

Info a!dTech

Web Development

TASK LIST



Task List

Task 1	Product landing page	Day 1-7
Task 2	CALCULATOR	Day 8-15
Task 3	Temperature Converter	Day 16-22
Task 4	Restaurant Website	Day 23-30

TASK 1:
Product landing page

- To develop a product landing page of a website, you must have sound knowledge of HTML and CSS.
- In this project, you will create columns and align the components of the landing page within the columns. You will have to perform basic editing tasks like cropping and resizing images, using design templates to make the layout more appealing, and so on.

TASK 2: CALCULATOR

- A basic calculator that performs simple arithmetic operations such as addition, subtraction, multiplication, and division. It takes two numerical inputs and an operator, and then calculates the result based on the chosen operator.
- The calculator is implemented using HTML, CSS, and JavaScript. The HTML defines the user interface, which consists of input fields for the two numbers and the operator, a button to trigger the calculation, and an output field to display the result.

TASK 3:
Temperature Converter

- A simple temperature converter website allows users to convert temperatures from one unit to another. It typically has an input field for the user to enter the temperature in a particular unit (such as Celsius or Fahrenheit), and buttons or radio buttons to select the desired output unit (such as Celsius, Fahrenheit, or Kelvin).
- The website is implemented using HTML, CSS, and JavaScript.

Steps to create the project:

1. Plan the program design and user interface, including the input and output fields and the conversion formula.
2. Use JavaScript to write the program code.
3. Create a function or method to perform the temperature conversion, based on the selected input and output scales.
4. Add error handling to handle invalid input, such as non-numeric values or unsupported temperature scales.
5. Implement a user interface to display the input and output fields and capture user input, using a graphical user interface (GUI) library or web framework, such as Java Swing, Python or JavaScript React.
6. Add a button or trigger to initiate the temperature conversion, and display the result in the output field.
7. Test the program for functionality, usability, and compatibility with different platforms and browsers.

TASK 4:
Restaurant Website

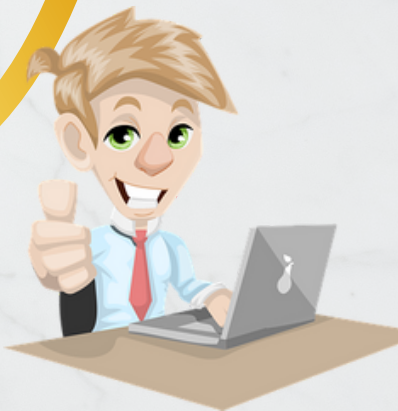
- This project aims to create a fully responsive restaurant website, you can add many pages and links to your website. You can extend the functionality by connecting it with a real-time database and allowing users to order food online.
- You must also deploy this using Netlify or GitHub Pages to showcase to the world that you are a great web developer.

Steps to create the project:

1. Plan the website layout, design, and content, including the homepage, menu page, location page, and contact page.
2. Use HTML to create the website structure and content, including headings, paragraphs, images, and links.
3. Use CSS to style the website, including fonts, colors, backgrounds, and layouts.
4. Use JavaScript to add interactivity to the website, such as navigation menus, dropdowns, and image sliders.
5. Use responsive design techniques to ensure the website is mobile-friendly and accessible on different devices.
6. Test the website for usability, functionality, and compatibility with different browsers and devices.

Info a!dTech

App Development



Task List

Task 1	Calculator App	Day 1-10
Task 2	To-Do List App	Day 11-20
Task 3	Weather App	Day 21-30

TASK 1:
Calculator App

Features:

- The app should have a clean and user-friendly interface.
- It should have buttons for numbers and operators (+, -, *, /).
- It should have a display area to show the calculation result.
- It should have a clear button to reset the calculation.
- It should be able to handle basic error handling like division by zero.

Steps to create the project:

1. Create a new Android Studio project and name it "Calculator App."
2. Design the user interface by adding buttons and a display area.
3. Implement the logic for each button, such as when the user clicks the number button, it should display on the screen.
4. Implement the logic for each operator button to perform the corresponding calculation.
5. Add the logic for the clear button to reset the calculation.
6. Implement error handling for division by zero.
7. Test the app on an emulator or a physical device.

TASK 2:
To-Do List App

Features:

- The app should have a clean and user-friendly interface.
- It should allow users to add new tasks to the list.
- It should allow users to mark tasks as complete.
- It should allow users to delete tasks from the list.
- It should have a display area to show the list of tasks.
- It should be able to persist the task data even when the app is closed.

Steps to create the project:

1. Create a new Android Studio project and name it "To-Do List App."
2. Design the user interface by adding a list view and buttons for adding, marking, and deleting tasks.
3. Create a custom adapter to display the list of tasks in the list view.
4. Implement the logic for adding new tasks to the list and updating the adapter.
5. Implement the logic for marking tasks as complete and updating the adapter.
6. Implement the logic for deleting tasks from the list and updating the adapter.
7. Use Shared Preferences or a local database to persist the task data even when the app is closed.
8. Test the app on an emulator or a physical device.

TASK 3:
Weather App

Features:

- The app should have a clean and user-friendly interface.
- It should allow users to search for a location by name or ZIP code.
- It should display the current weather conditions, including temperature, humidity, wind speed, and precipitation.
- It should display the forecast for the next few days.
- It should be able to handle basic error handling like invalid location.

Steps to create the project:

1. Create a new Android Studio project and name it "Weather App."
2. Design the user interface by adding input fields, buttons, and display areas for weather data.
3. Use an API service like OpenWeatherMap to fetch weather data for the given location.
4. Implement the logic for parsing the weather data and displaying it on the screen.
5. Implement the logic for handling errors and displaying error messages to the user.
6. Test the app on an emulator or a physical device.

Info a!dTech

Python

Task List

Task 1	Number Guessing Game	Day 1-7
Task 2	To-Do List App	Day 8-15
Task 3	Dice Roller	Day 16-22
Task 4	Weather App	Day 23-30

TASK 1:
Number Guessing Game

Features:

- The game should have a clean and user-friendly interface.
- It should allow users to input their name and start the game.
- It should generate a random number between 1 and 100.
- It should allow users to guess the number within 10 attempts.
- It should provide feedback to the user after each guess, letting them know if their guess was too high or too low.
- It should declare the user as the winner if they guess the number correctly.
- It should give the user the option to play again or quit the game.

Steps to create the project:

1. Open a new Python file in your favorite text editor or IDE.
2. Write a welcome message to the user, asking them for their name and explaining the rules of the game.
3. Generate a random number between 1 and 100 using the random module.
4. Use a loop to allow the user to input their guess within 10 attempts.
5. Compare the user's guess with the random number and provide feedback to the user, letting them know if their guess was too high or too low.
6. If the user guesses the number correctly within the 10 attempts, declare them the winner and ask if they want to play again.
7. If the user doesn't guess the number correctly within the 10 attempts, declare the game over and ask if they want to play again.
8. Test the game by running the Python file in your terminal or console.

TASK 2:
To-Do List App

Create a simple to-do list app where the user can add, delete, and view their tasks.

Features:

- The app should have a clean and user-friendly interface.
- It should allow users to add a new task to their list.
- It should allow users to delete a task from their list.
- It should allow users to view their current list of tasks.
- It should allow users to save their list of tasks to a file and load it back again later.

Steps to create the project:

1. Open a new Python file in your favourite text editor or IDE.
2. Define a class Task with attributes such as title, description, and status.
3. Define a class ToDoList with methods such as add_task(), delete_task(), view_tasks(), save_tasks(), and load_tasks().
4. Implement the logic for adding a new task to the to-do list, deleting a task from the to-do list, and viewing the current list of tasks.
5. Implement the logic for saving the list of tasks to a file and loading it back again later.
6. Test the app by running the Python file in your terminal or console.

TASK 3:
Dice Roller

Create a simple dice rolling app that simulates rolling a pair of dice.

Features:

- The app should have a clean and user-friendly interface.
- It should allow users to select the number of dice they want to roll.
- It should simulate rolling the dice and display the result.
- It should allow users to roll the dice again or quit the app.

Steps to create the project:

1. Open a new Python file in your favourite text editor or IDE.
2. Define a function roll_dice() that simulates rolling a pair of dice.
3. Use the random module to generate random numbers between 1 and 6 to simulate the dice rolls.
4. Implement the logic for allowing the user to select the number of dice they want to roll.
5. Implement the logic for displaying the result of the dice rolls.
6. Allow users to roll the dice again or quit the app.
7. Test the app by running the Python file in your terminal or console.

TASK 4:
Weather App

Features:

- The app should have a clean and user-friendly interface.
- It should allow users to input a location and fetch the current weather data.
- It should display the temperature, humidity, wind speed, and weather condition of the location.
- It should provide an option for the user to view the forecast for the next few days.
- It should allow users to save their favourite locations and view their weather data easily.

Steps to create the project:

1. Open a new Python file in your favourite text editor or IDE.
2. Use an API service like OpenWeatherMap to fetch the weather data of a given location.
3. Implement the logic for parsing the weather data and displaying it on the screen.
4. Use a loop to allow the user to input a location and fetch its weather data.
5. Allow users to view the forecast for the next few days.
6. Allow users to save their favourite locations and view their weather data easily.
7. Test the app by running the Python file in your terminal or console.

Info a!dTech

Machine learning

Task List

Task 1	Iris Flower Classification Dataset: Here	Day 1-10
Task 2	Movie Recommendation System Dataset: Here	Day 11-20
Task 3	Credit Card Fraud Detection Dataset: Here	Day 21-30

TASK 1:
Iris Flower Classification

Create a machine learning model that can classify the species of an iris flower based on its sepal and petal length and width.

Dataset: Here

Steps to create the project:

1. Load the Iris dataset into your Python environment. You can use a library like Scikit-learn to load the dataset.
2. Pre-process the dataset by splitting it into training and testing sets.
3. Explore the dataset by visualizing the data using scatterplots or histograms.
4. Select a machine learning algorithm to train your model. You can start with a simple algorithm like K-Nearest Neighbours or Decision Trees.
5. Train your model using the training set.
6. Evaluate your model's performance on the testing set.
7. Use your model to make predictions on new data.
8. Test your model by inputting new values for sepal length, sepal width, petal length, and petal width to see the predicted species of iris flower.

TASK 2:
Movie Recommendation System

Create a machine learning model that can recommend movies to users based on their preferences.

Dataset: Here

Steps to create the project:

1. Load the MovieLens dataset into your Python environment. You can use a library like Pandas to load the dataset.
2. Pre-process the dataset by cleaning the data, removing duplicates and missing values, and converting the data into a matrix format.
3. Split the dataset into training and testing sets.
4. Implement a collaborative filtering algorithm to train your model. Collaborative filtering is a technique that recommends movies to users based on their ratings and the ratings of similar users.
5. Train your model using the training set.
6. Evaluate your model's performance on the testing set using metrics like Mean Absolute Error or Root Mean Squared Error.
7. Use your model to make movie recommendations for users based on their preferences.
8. Test your model by inputting new user ratings to see the recommended movies.

TASK 3:
Credit Card Fraud Detection

Dataset: Here

Steps to create the project:

1. Load the Credit Card Fraud Detection dataset into your Python environment. You can use a library like Pandas to load the dataset.
2. Pre-process the dataset by scaling the features and handling any missing values or outliers.
3. Split the dataset into training and testing sets.
4. Implement a classification algorithm to train your model. You can start with a simple algorithm like Logistic Regression or Support Vector Machines.
5. Train your model using the training set.
6. Evaluate your model's performance on the testing set using metrics like accuracy, precision, recall, and F1 score.
7. Use your model to predict whether a new credit card transaction is fraudulent or not.
8. Test your model by inputting new values for the features to see the predicted outcome.

Info a!dTech

JAVA

Task List

Task 1	Create a password generator	Day 1-7
Task 2	Temperature Conversion Utility	Day 8-15
Task 3	Student Record Management System	Day 16-30

TASK 1:
Create a Password Generator

Steps to create the project:

1. Define the length of the password: Decide on the length of the password that you want to generate.
2. Define the character set: Decide on the character set you want to use to generate the password. This can include uppercase and lowercase letters, numbers, and special characters.
3. Create a function to generate the password: Write a function that takes in the length of the password and the character set as parameters and generates a random password.
4. Use random number generation: Use Java's built-in random number generator to generate random characters from the chosen character set.
5. Convert the password to a string: Convert the generated password to a string and return it from the function.
6. Add user input: Add functionality to allow the user to input the desired length of the password and the character set to be used.
7. Test the program: Test the program by generating passwords of different lengths and using different character sets.

TASK 2:
Temperature Conversion

Create a program that can convert temperature between Celsius, Fahrenheit, and Kelvin scales.

Steps to create the project:

1. Create a user interface using Java Swing or JavaFX to display the temperature conversion utility and capture user input.
2. Implement methods to convert temperature between Celsius, Fahrenheit, and Kelvin scales.
3. Use exception handling to handle errors, such as entering invalid input.
4. Add a button to clear the utility display and reset the conversion.
5. Add a button to switch between Celsius, Fahrenheit, and Kelvin scales.

TASK 3:
Student Record Management System

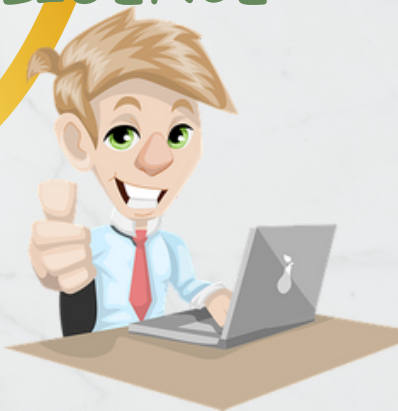
Create a program that can manage student records, including adding new students, removing students, and displaying student information.

Steps to create the project:

1. Define a Student class that contains information about the student, such as name, roll number, address, and phone number.
2. Implement a StudentRecord class that contains an array or list of Student objects.
3. Create methods for adding a new student to the record, removing a student from the record, and displaying student information based on the roll number or name.
4. Implement a user interface using Java Swing or JavaFX to allow users to interact with the student record system.
5. Save the student record to a file and load it back when the program starts.

Info a!dTech

ARTIFICIAL INTELLIGENCE



Task List

Task 1	Chatbot	Day 1-10
Task 2	Voice Recognition	Day 11-20
Task 3	Sentiment Analysis	Day 21-30

TASK 1:
Chatbot

Description: Create a chatbot that can interact with users and provide helpful responses to their queries.

Steps to create the project:

1. Plan the chatbot design and functionality, including the user interface and the natural language processing (NLP) algorithm.
2. Use a programming language of your choice, such as Python, to write the program code.
3. Implement an NLP library or framework, such as Natural Language Toolkit (NLTK), to process and analyse user input and generate appropriate responses.
4. Train the chatbot using a dataset of sample conversations, to improve its accuracy and relevance.
5. Implement a user interface, such as a web-based chat window or a messaging platform, to allow users to interact with the chatbot.
6. Add error handling to handle invalid input, such as unrecognized queries or network errors.
7. Test the chatbot for functionality, usability, and performance, using different test scenarios and feedback from users.

TASK 2:
Voice Recognition

Info a!dTech

Graphics & Logo Design

TASK LIST

Task List

Task 1	Logo design for Info aidTech (Color: #006B38 with other combination)
Task 2	Food Menu
Task 3	Book cover design
Task 4	Portfolio website Design

TASK 1:
Logo design for Info aidTech

1. Before you begin designing the logo, you need to understand the brand and its values. Info aidTech is a technology company, so it is important to create a logo that reflects innovation, technology, and reliability.
2. Choose a color scheme that reflects the brand's values and personality. Blue and green are common colors for technology companies, as they represent trust and growth, respectively.
3. Once you have some initial concepts, refine the design by creating digital versions of the logo. Use a vector-based software such as Adobe Illustrator to ensure that the logo is scalable and can be used in different sizes and formats.

Note:

TASK 2:
Food Menu

Choose a restaurant and design a new menu for them. This can include designing a layout, typography, and incorporating relevant images or graphics.

Info aidTech

If you haven't done so yet, please support us.
With your help, we can reach out to more students

Connect With US

Info aidTech

Info aidTech

Info aidTech

Info_aidTech

internshipinfoaidtech@gmail.com