Team Name

200 NOT OK

Team Details

Team Leader: Devesh Sharma
Team Member: Mayank
Team Member: Ayush Patel
Team Member: Navkiran Kaur
Team Member: Gaurav Prasher

Guru Nanak Dev Engineering College
Ludhiana, Punjab

instructo

Problem Statement

In today’s fast-evolving tech landscape, aspiring programmers struggle with fragmented learning resources, lack of personalized guidance, and limited real-time feedback. Traditional coding platforms often provide direct solutions rather than fostering deep problem-solving skills, leaving learners dependent rather than independent.

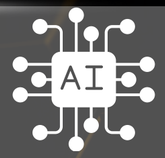
Proposed Solution

Instructo revolutionizes coding education by using an AI mentor to guide learners through problem-solving, rather than providing direct answers. This approach fosters independent thinking and prepares users for real-world industry challenges through personalized, project-based learning.

- AI-guided problem-solving eliminates answer-based learning.
- Personalized learning paths and project-based modules allow self-paced progression.
- Gamified tracking and community collaboration enhance engagement and motivation.

Features

Smart AI Mentor



Our AI never provides direct code, instead, it guides users with logical explanations, enhancing problem-solving skills and reducing copy-pasting habits.

Intelligent Code Validation

Beyond runtime, AI validates code logic, detecting and rejecting incorrect approaches such as hardcoded patterns.

Project-Based Learning

Master coding by building real-world project components like user authentication in any language, ensuring practical understanding with step by step guidance

Interactive Challenge Modes

- Output-Based Coding – Write the code to match an AI given output.
- Complete the Code – Users fix and finish an incomplete code snippet.

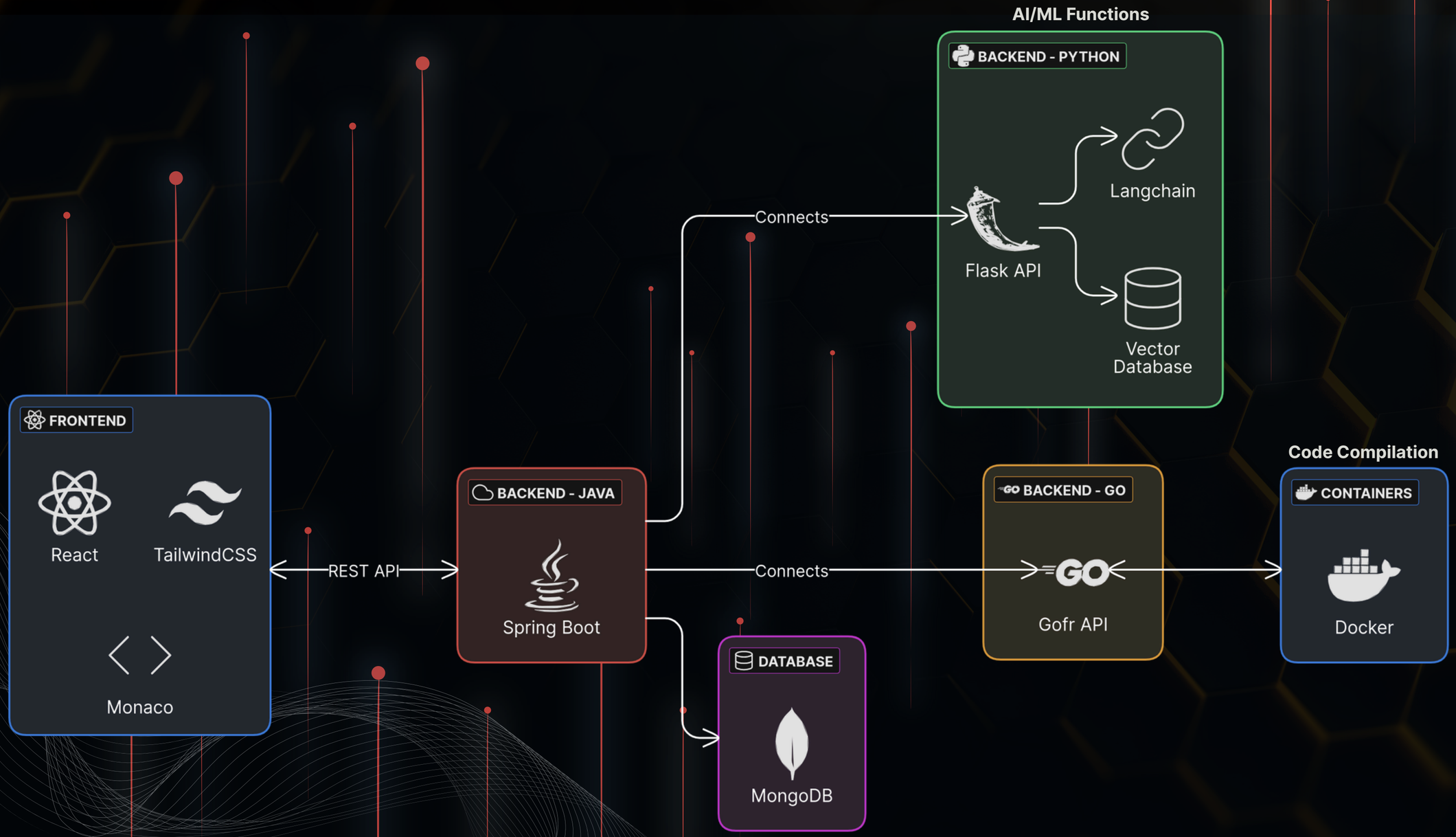
Gamified Learning & Progress Tracking

Learners begin their language journey with a tailored starting point via an AI proficiency test, and continue with progress tracking and structured roadmaps throughout each course

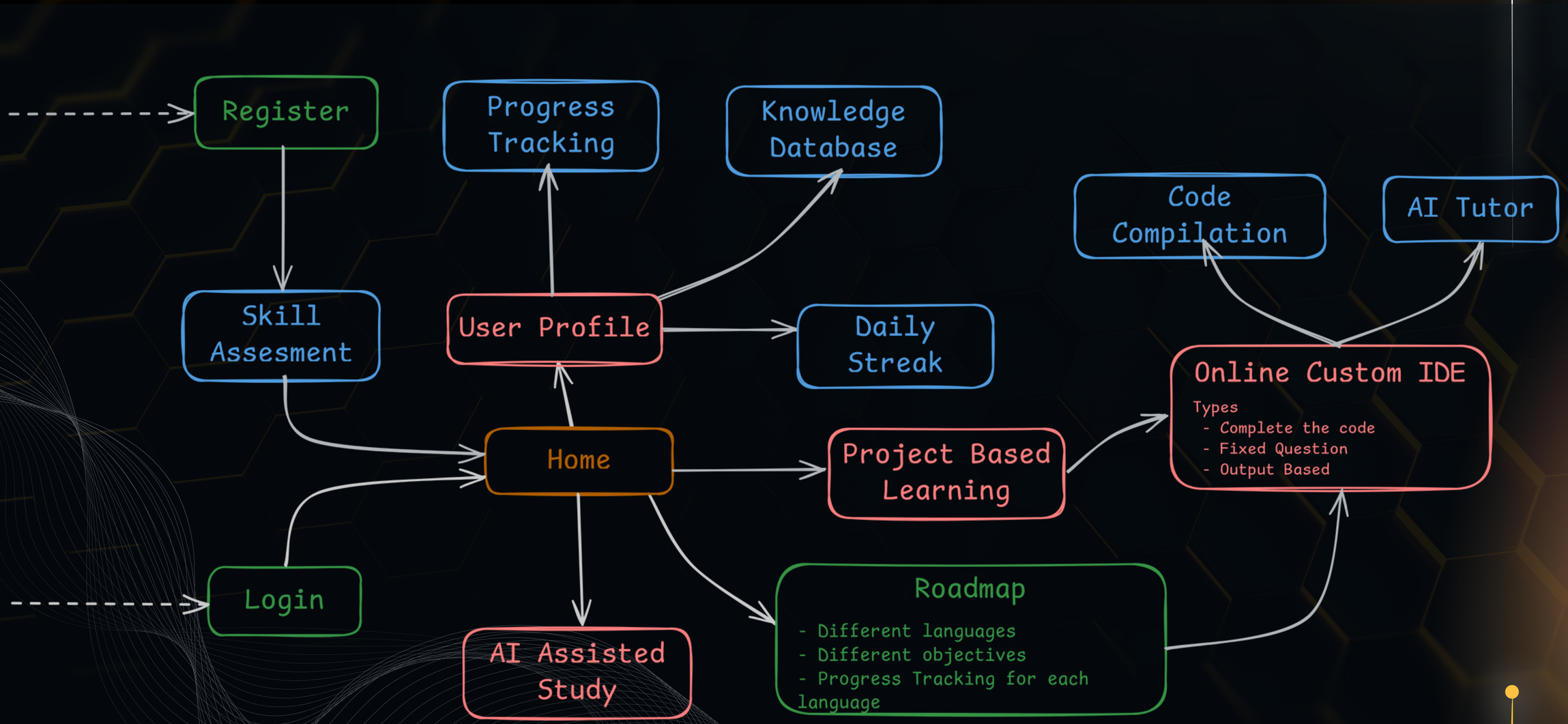
Bonus Feature: Explain Like I’m 5

A unique mode that breaks down complex coding concepts into super simple explanations, making learning fun and effortless.

Tech Stack


AI/ML Functions
BACKEND - PYTHON
Langchain
Flask API
Vector Database
Connects
FRONTEND
React
TailwindCSS
Monaco
REST API
BACKEND - JAVA
Spring Boot
Connects
BACKEND - GO
Gofr API
Code Compilation
CONTAINERS
Docker
DATABASE
MongoDB

Workflow Diagram



Target Audience

- Aspiring Developers & Students
- Self-Learners & Hobbyists
- Universities & Coding Bootcamps
- Tech Professionals

Engagement

Interactive AI Coaching
Offers instant, personalized feedback, acting as a patient mentor to transform coding challenges into engaging problem-solving experiences instead of frustrating obstacles.

Adaptive Difficulty Scaling
Adjusts challenges automatically to keep you in the optimal learning zone, avoiding boredom and overwhelm.

Project-Based Learning
Involves creating personalized projects with achievable steps and milestones, fostering a continuous sense of accomplishment and momentum.

Gamified Learning Experience
Earn XP, unlock achievements, and climb skill levels, transforming coding practice into an exciting personal challenge.

Future Scope

Certification Programs

Develop industry-recognized certifications based on project completion and skill mastery, providing tangible credentials for job seekers.

Mobile Learning Experience

Extend the platform to mobile devices with a specialized app that enables on-the-go learning through bite-sized coding challenges and progress tracking.

Collaborative Coding Environment

Create real-time pair programming capabilities where users can collaborate on projects together, simulating team development environments.

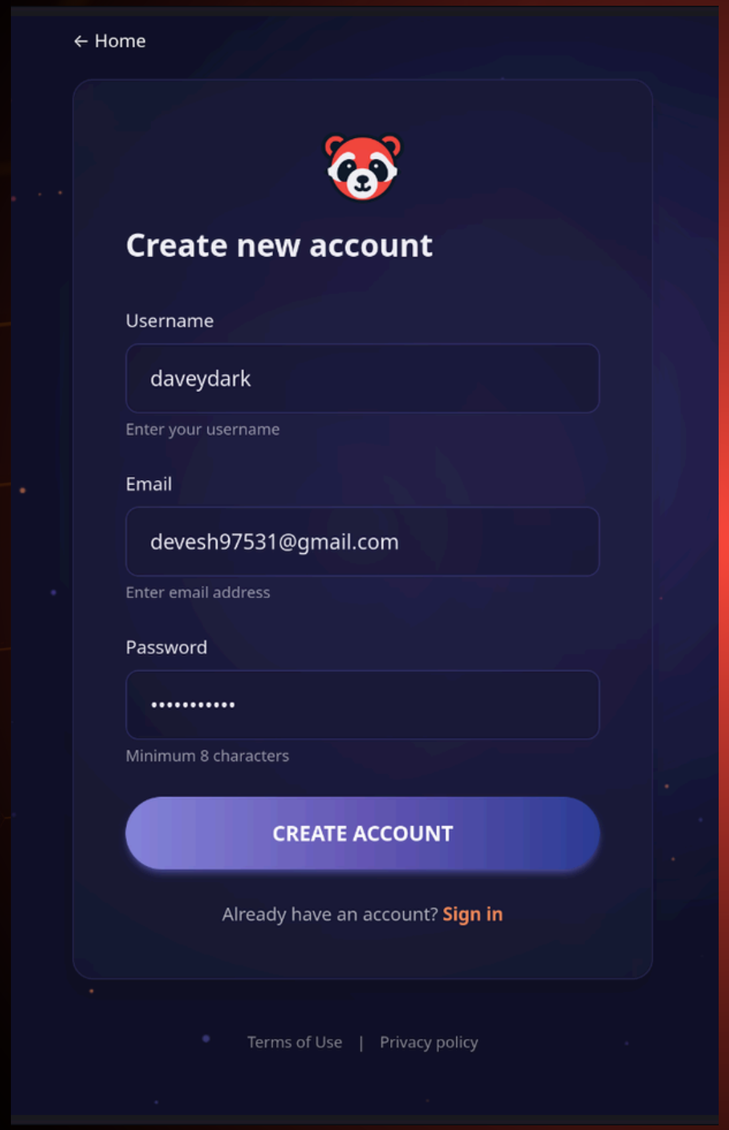
Enterprise Learning Management

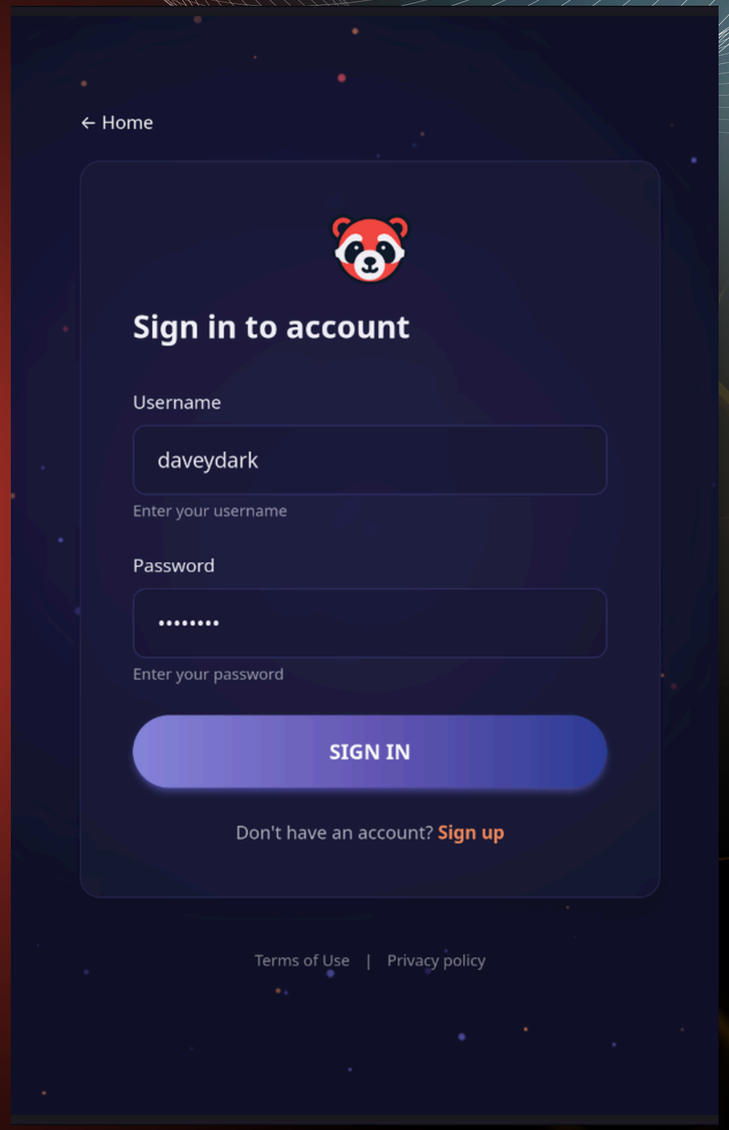
Expand to corporate environments with team progress tracking, skill gap analysis, and customized learning paths aligned with specific organizational technology stacks.

Home Page


Code Editor
instructo
Sign Up
AI Powered Coding Tutor
Instructo revolutionizes coding education by using an AI mentor to guide learners through problem solving rathar than providing direct answers
Get Started
Learn More
Smart AI Mentor
Our AI never provides direct code, instead, it guides users with logical
Intelligent Code Validation
Beyond runtime, AI validates code logic,
Interactive Challenge Modes
Output-Based Coding – Write the code to match an

Sign Up/ Sign In





Language Selection

Quiz Page

Python Quiz

Question 1

What is the output of `print(2 + '2')`?

- 4
- 22
- TypeError
- None

Question 2

Which of the following is NOT a valid data type in Python?

- int
- float
- string

Question 4

What is the output of `print(5 // 2)`?

- 2.5
- 2 × Incorrect
- 2.0
- 3

Explanation: In Python, lists (and other sequences) are zero-indexed, meaning the first element is at index 0.

Question 5

Which keyword is used to define a function in Python?

- function
- def ✓ Correct
- func
- define

Roadmaps


Your python Learning Roadmap
Follow this step-by-step roadmap to master python. Click on any step to view details and resources.
Variables & Data Types
Control Structures
Functions & Methods
Data Structures
Object-Oriented Programming
Advanced Concepts
Variables & Data Types
Completed • Step 1 of 6
Practice
Objective
Foundation of Programming
Description
Learn how to declare variables and understand different data types in python.
Complete the Code
Output Based Challenge
Problem Based Challenge
Next Step
© 2025 Instructo Platform • Learning Paths for All Skill Levels

Practice

Code Editor

```
1  let a = 10;
2  const b = 20;
3  var c = 30;
4  console.log(a);
5  console.log(b);
6  console.log(c);
```

Run Code

Output

```
10
20
30
```

Objective

Practice: Variables & Data Types - Foundation of Programming

Description

Learn how to declare variables and understand different data types in javascript. This is a free practice session. You can code anything related to the objective above.

Potential Improvements

• Add comments explaining the purpose of the code. • Use more descriptive variable names. • Consider using

```
const
```

for all variables that don't need reassignment.

Detailed Review

Structure/Organization

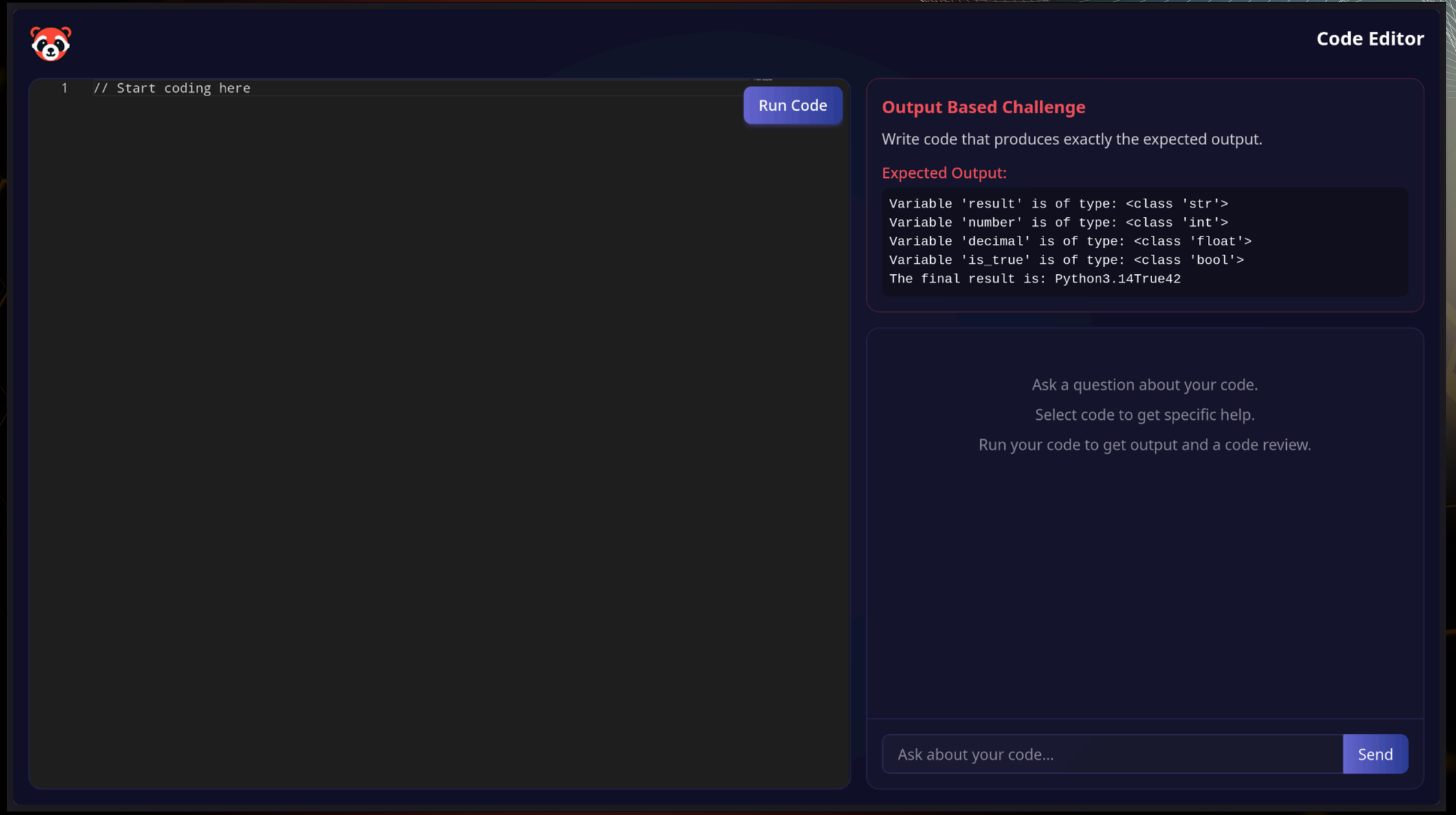
Observations: • The code is very simple and lacks context. It's difficult to understand its purpose without more information. • There's no apparent structure or organization beyond simple variable declarations and console logs.

Suggestions: • Add a comment at the beginning of the file explaining what the code is supposed to do. • If this is part of a larger program, consider encapsulating this code within a function or module to improve organization.

Ask about your code...

Send

Output Based Code



Complete the code

Code Editor

```
1  # Variable declaration and data types
2
3  # Integer
4  age = ____________________
5  print("Age:", age)
6
7  # Float
8  price = ____________________
9  print("Price:", price)
10
11 # String
12 name = ____________________
13 print("Name:", name)
14
15 # Boolean
16 is_student = ____________________
17 print("Is student:", is_student)
18
19 # Demonstrating type conversion
20 # Convert age to float
21 age_float = ________(age)
22 print("Age as float:", age_float)
23
24 # String concatenation
25 message = "Hello, " + name + "! You are " + ________(age) + " years old."
```

Run Code

Output ×

```
Age: 25 <class 'int'>
Height: 5.9 <class 'float'>
Name: Alice <class 'str'>
Is student: True <class 'bool'>
Numbers: [1, 2, 3, 4, 5] <class 'list'>
Coordinates: (10, 20) <class 'tuple'>
Student: {'name': 'Bob', 'age': 20, 'major': 'Computer Science'} <class 'dict'>
Height as integer: 5 <class 'int'>
Hello, my name is Alice and I am 25 years old.
Hello, my name is Alice and I am 25 years old
```

Code Completion Challenge

Your task is to complete the code by filling in the missing parts.

Observations: • Variable names are generally acceptable but could be more descriptive in some cases. • The code demonstrates basic data types and type conversion, which is good for beginners. • Using f-strings is a good practice for modern Python development.

Suggestions: • For variables like

```
height
```

, consider

```
height_in_meters
```

for clarity. • When demonstrating data types, also include examples of immutability (e.g., try modifying a tuple). • Always include error handling when dealing with user inputs or external data sources to avoid unexpected crashes.

Learning Resources

• Python Data Types • String Formatting in Python • Python Lists

10:19:25 am

Ask about your code... Send

Complete the code

Code Editor

```
1  # Variable declaration and data types
2
3  # Integer
4  age = ____________________
5  print("Age:", age)
6
7  # Float
8  price = ____________________
9  print("Price:", price)
10
11 # String
12 name = ____________________
13 print("Name:", name)
14
15 # Boolean
16 is_student = ____________________
17 print("Is student:", is_student)
18
19 # Demonstrating type conversion
20 # Convert age to float
21 age_float = ________(age)
22 print("Age as float:", age_float)
23
24 # String concatenation
25 message = "Hello, " + name + "! You are " + ________(age) + " years old."
```

Run Code

Output ×

```
Age: 25 <class 'int'>
Height: 5.9 <class 'float'>
Name: Alice <class 'str'>
Is student: True <class 'bool'>
Numbers: [1, 2, 3, 4, 5] <class 'list'>
Coordinates: (10, 20) <class 'tuple'>
Student: {'name': 'Bob', 'age': 20, 'major': 'Computer Science'} <class 'dict'>
Height as integer: 5 <class 'int'>
Hello, my name is Alice and I am 25 years old.
Hello, my name is Alice and I am 25 years old
```

Code Completion Challenge

Your task is to complete the code by filling in the missing parts.

Observations: • Variable names are generally acceptable but could be more descriptive in some cases. • The code demonstrates basic data types and type conversion, which is good for beginners. • Using f-strings is a good practice for modern Python development.

Suggestions: • For variables like

```
height
```

, consider

```
height_in_meters
```

for clarity. • When demonstrating data types, also include examples of immutability (e.g., try modifying a tuple). • Always include error handling when dealing with user inputs or external data sources to avoid unexpected crashes.

Learning Resources

• Python Data Types • String Formatting in Python • Python Lists

10:19:25 am

Ask about your code... Send

Problem Based Challenge