CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

cspit
Chandubhai S Patel Institute of Technology

Faculty of Technology and Engineering

Chandubhai S. Patel Institute of Technology

Date: / /

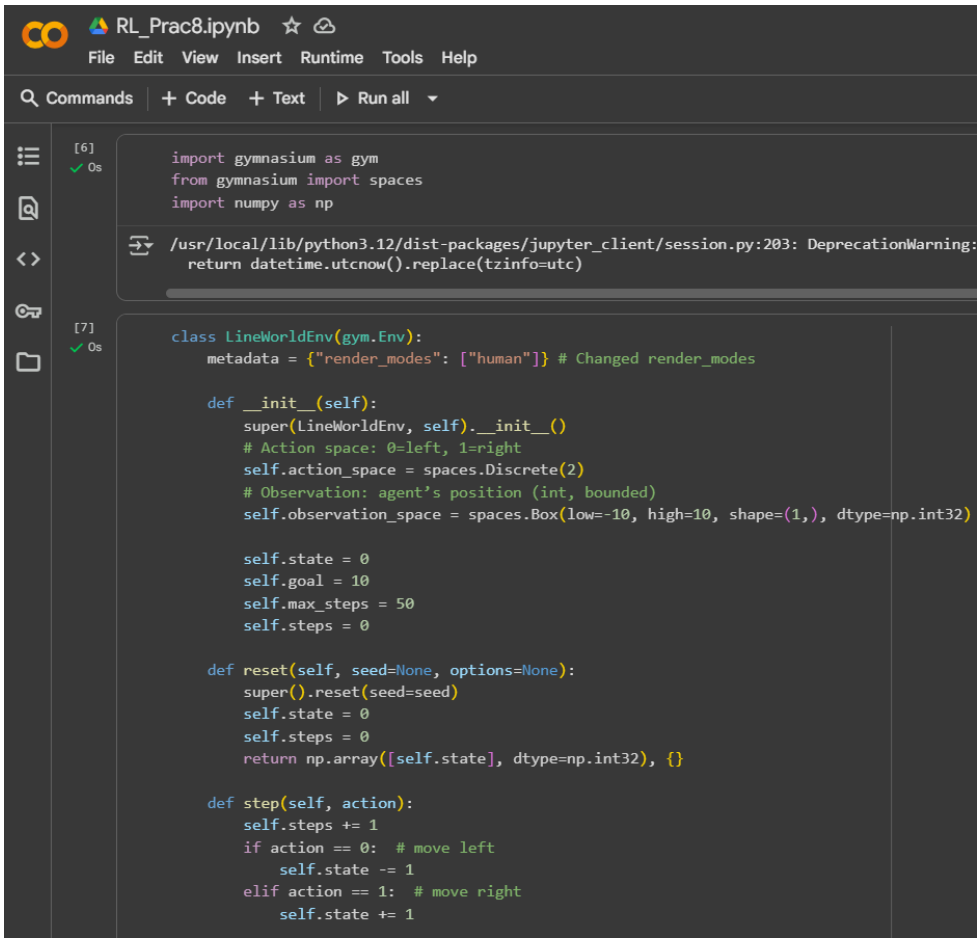
Practical Performa

Academic Year	:	2025-26	Semester	:	7th
Course code	:	OCCSE4001	Course name	:	Reinforcement Learning

Practical- No. 8

Aim: Design and register a custom Gym environment from scratch, understand its structure, and train/test RL agents on custom tasks.

Code:



RL_Prac8.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[7] ✓ 0s

```
            # Reward logic
            if self.state == self.goal:
                reward = 1.0
                terminated = True
            else:
                reward = -0.1
                terminated = False

            truncated = self.steps >= self.max_steps
            return np.array([self.state], dtype=np.int32), reward, terminated, truncated, {}

        def render(self):
            print(f"Position: {self.state}")

        def close(self):
            pass
```

[8] ✓ 0s

```
from gymnasium.envs.registration import register

register(
    id="LineWorld-v0",
    entry_point=LineWorldEnv,
)
```

RL_Prac8.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[9] ✓ 0s

```
env = gym.make("LineWorld-v0")

state, _ = env.reset()
print("Initial state:", state)

for _ in range(10):
    action = env.action_space.sample()  # random agent
    # Gymnasium's step returns 5 values: observation, reward, terminated, truncated, info
    next_state, reward, terminated, truncated, _ = env.step(action)
    print(f"Action: {action}, State: {next_state}, Reward: {reward}")
    if terminated or truncated:
        print("Episode finished")
        break
```

RL_Prac8.ipynb

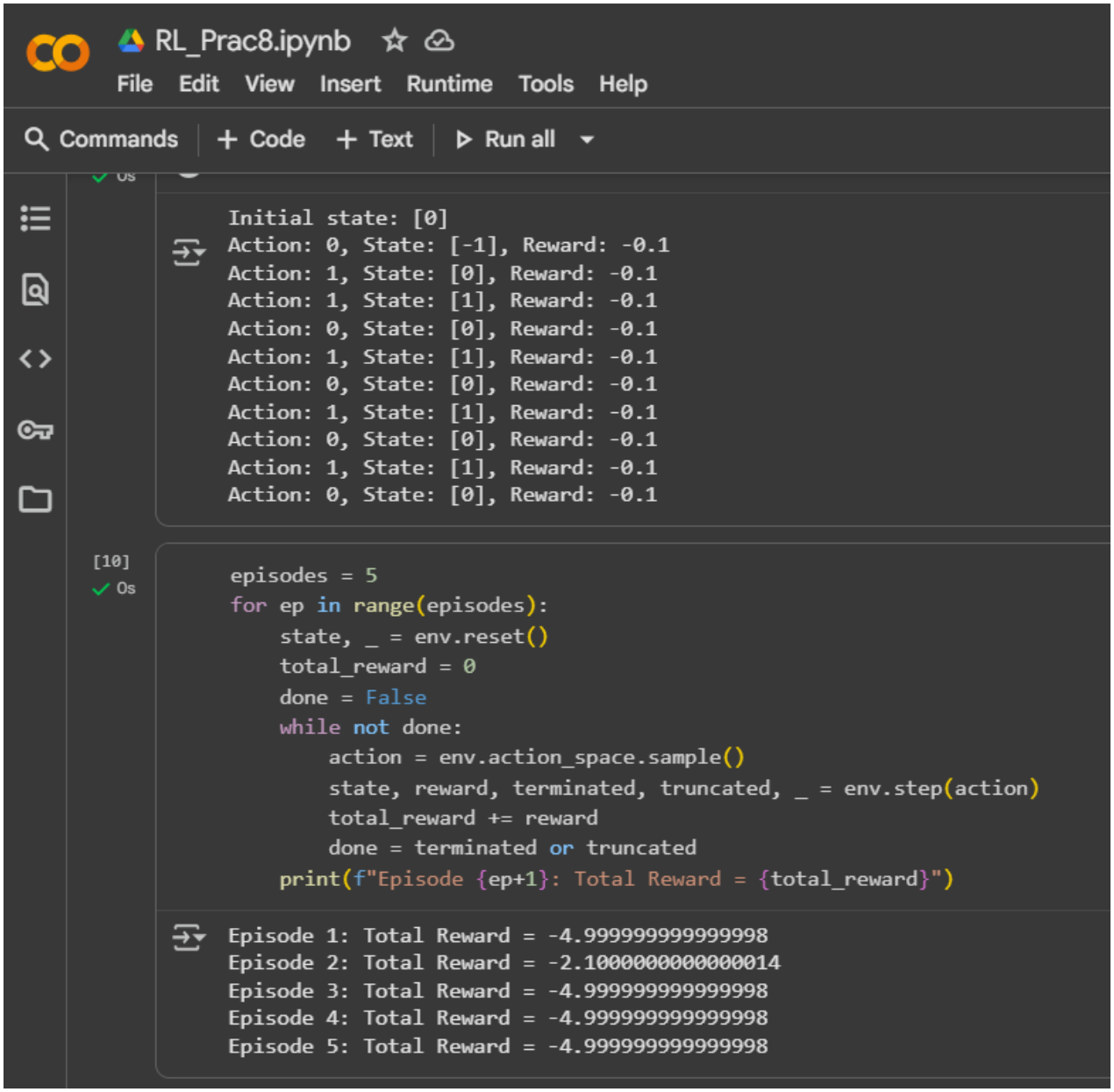
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[10] ✓ 0s

```
episodes = 5
for ep in range(episodes):
    state, _ = env.reset()
    total_reward = 0
    done = False
    while not done:
        action = env.action_space.sample()
        state, reward, terminated, truncated, _ = env.step(action)
        total_reward += reward
        done = terminated or truncated
    print(f"Episode {ep+1}: Total Reward = {total_reward}")
```

Output:



Grade/Marks **Sign of Lab Teacher with Date**

(_____ / 10)