

Great – here's a Unity-specific version you can submit. It's clean, concise, and formatted properly for a 1-page document.

---

# Mini Development Report

## Step 1 – Project Overview

**Project Name:** Neon Runner

**Genre / Type:** 2D Endless Runner

**Platform:** PC (Windows)

**Team Size:** Solo Developer

**Duration:** 3 Weeks

### Project Goal:

Neon Runner is a fast-paced 2D endless runner created in Unity. The goal of the project was to practice core Unity skills such as player movement, physics, collision detection, UI systems, and basic game optimization.

---

## Step 2 – Development Report

The game was developed using Unity (URP) and C#. Visual Studio was used for scripting. I started by creating a basic prototype focusing only on player movement and jumping using Unity's Rigidbody2D physics system. After testing the core mechanic, I implemented obstacle spawning, a scoring system, and increasing game speed over time. UI elements such as the start menu, score display, and game over screen were created using Unity's Canvas system.

---

## Step 3 – Problems Encountered

1. Player jump physics felt too floaty and not responsive enough.
  2. Obstacles sometimes spawned too close together, making the game unfair.
  3. Performance decreased slightly due to frequent object instantiation and destruction.
-

## Step 4 – Solutions / Adjustments

1. I adjusted gravity scale and jump force values in Rigidbody2D to improve responsiveness.
  2. I implemented a minimum spawn distance and random spawn timing control.
  3. I used an object pooling system to reuse obstacles instead of constantly creating and destroying them.
- 

## Step 5 – Key Learning

I learned that polishing core mechanics early is essential before adding new features. Performance optimization techniques like object pooling are important even in small projects.