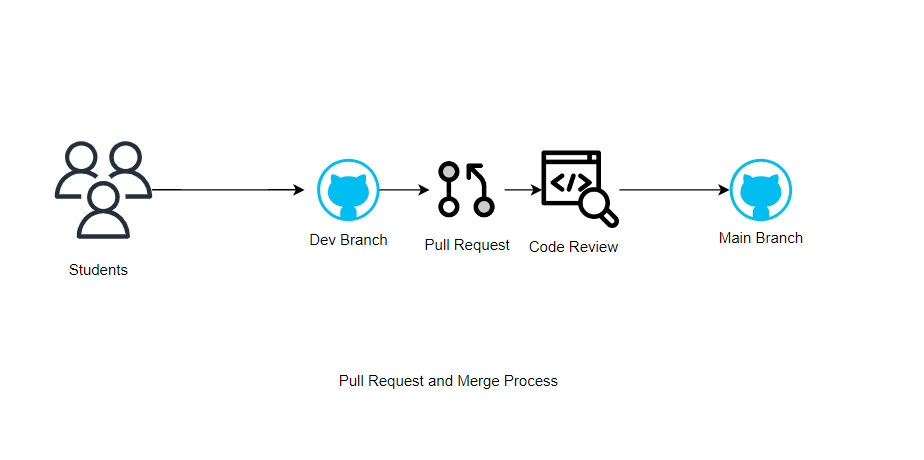
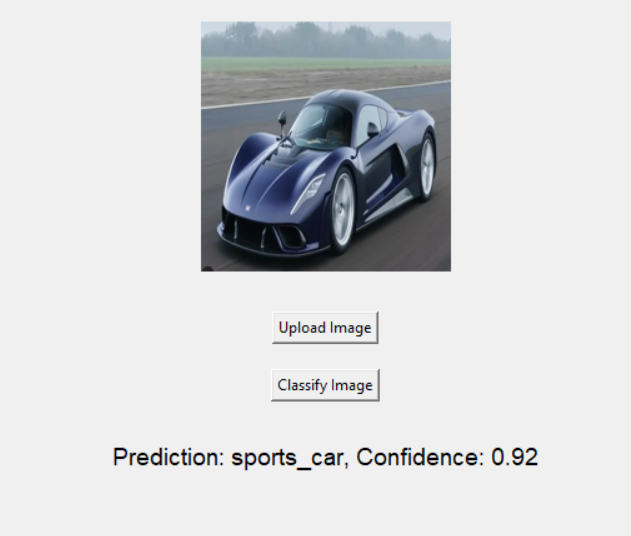
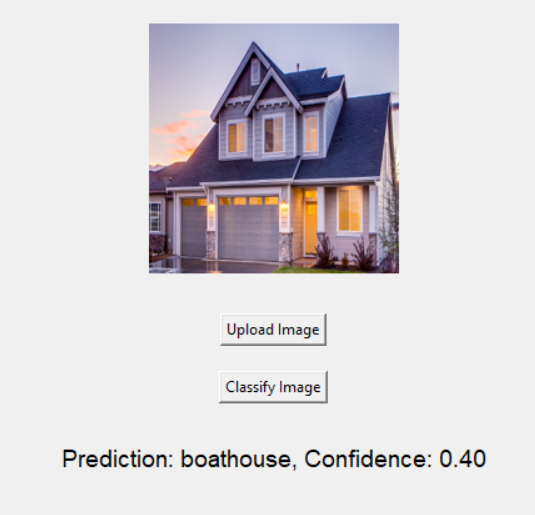
Here's a diagram showing the process where students first push their code to a branch, followed by a code review, and finally merge into the main branch. This workflow ensures that changes are reviewed before being integrated into the main project. [](https://private-user-images.githubusercontent.com/62823467/367557548-ee8569fc-baa9-4354-9fcb-97d48b8edc88.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..5jfLRJJnwWrNjqDAu2N6qiiXh4LoFumvRrF8IwLAdFw)









**Question 2**

**SideScrolling 2D Game**

**Game Overview:**

The game involves a player character with basic movements (run, jump, shoot projectiles), enemies to defeat, collectibles, and a camera that follows the player. The design includes 3 levels with a boss enemy at the end of each. The player has health and lives; the score is based on enemy defeats and collected items.

**Game Features Breakdown:**

**1. Player Class**

**Attributes:**

* Speed for running.
* Jump height for upward movement.
* Health and lives system.

**Methods:**

* Control player movements: `run()`, `jump()`.
* Player's ability to shoot projectiles: `shoot()`.
* Take damage and update health.

**2. Projectile Class**

**Attributes:**

* Speed and direction of the projectile.
* Damage to be inflicted on enemies.

**Methods:**

* Movement of the projectile: `move()`.
* Collision detection with enemies: `check\_collision()`.

**3. Enemy Class**

* Placeholder for enemy behavior, like movement patterns and attack mechanisms.
* Health and damage dealing logic.
* Special boss enemy at the end of each level.

**4. Collectible Class**

* Types: Health boosts, extra lives, ammunition.
* Each collectible affects the player’s attributes (e.g., more health or ammo)

**5. Level Design**

* 3 distinct levels with progressively harder enemies and challenges.
* Each level ends with a boss fight.
* The layout includes obstacles like water, mountains, and trees to make the environment visually appealing.

**6. Scoring System**

* Score increases based on:
* Enemies defeated.
* Collectibles picked up.

**7. Health Bar**

* Display both the player’s health and enemy health on the screen.

**8. Game Over Screen**

* When the player’s lives drop to zero, the game will show a Game Over screen with an option to restart.

**9. Bonus: Dynamic Camera**

* The camera should smoothly follow the player to keep them centered in the middle of the screen, creating a fluid gameplay experience.

**Game Controls:**

* A and D for movement left and right.
* W to jump.
* Space to shoot projectiles.

**Installation and Usage:**

**1. Clone the repository:**

git clone **https://github.com/Bipuojha1/softwarenow\_assignment3.git**

**2. Install the required dependencies**

pip install pygame

**3. Run the game**

python main.py

The game uses simple keys to control the character, making it user-friendly, and its dynamic elements like camera movement, smooth scrolling, and various environments create an engaging experience.

**Output Images**

