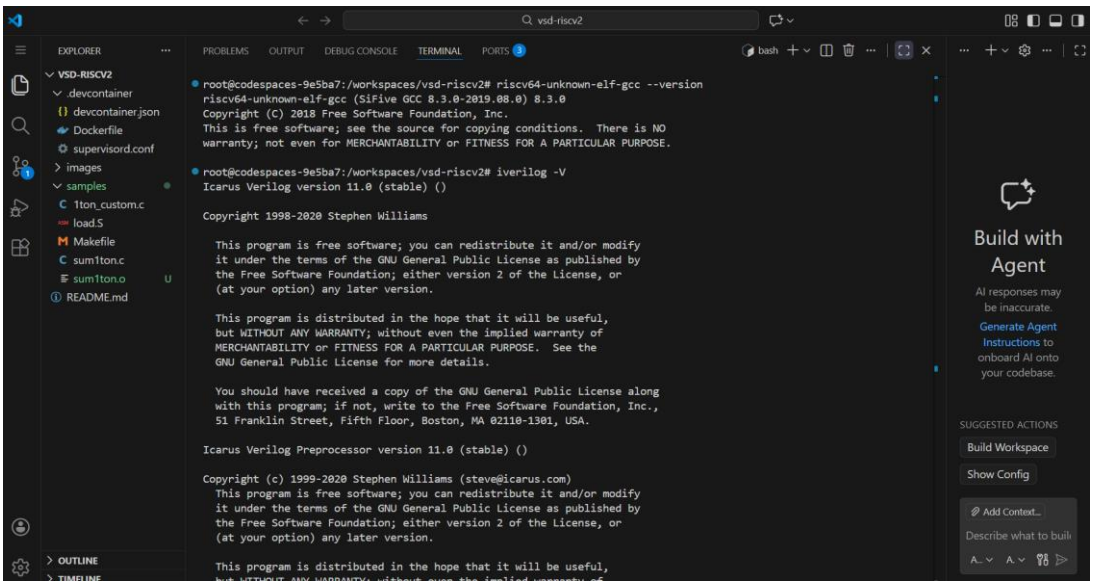
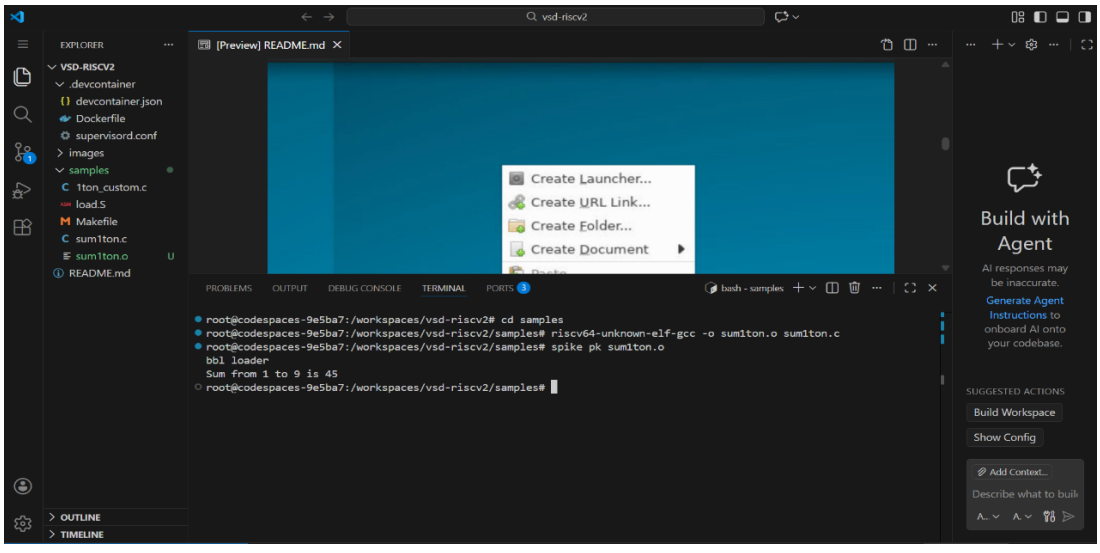
On Day 1, a cloud-based RISC-V development environment was set up using GitHub Codespaces. The RISC-V GCC toolchain, Spike simulator, and Icarus Verilog were verified, and a sample C program was successfully compiled and executed on both the native system and the RISC-V simulator. A GUI Linux desktop via noVNC was also used for editing and running programs, establishing the foundation for further RISC-V and hardware development

1.A new GitHub Codespace was created, the environment was built, and the RISC-V toolchain setup was verified.

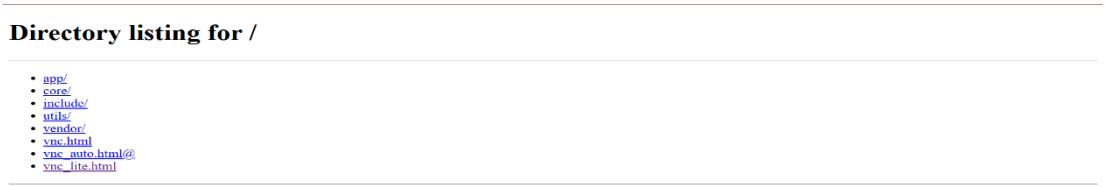
→ riscv64-unknown-elf-gcc –version
→ spike --version
→ iverilog -V



2. A sample C program was compiled using the RISC-V GCC cross-compiler and executed on the Spike simulator inside the Codespace, confirming that the RISC-V development environment was functioning correctly.



3. Opened the Linux desktop interface through Codespaces by navigating to the **PORTS** tab and accessing the forwarded **noVNC Desktop (6080)** address in the browser.



4. Linux Desktop Window A new browser tab opens showing the Linux GUI environment after clicking on the **vnc_lite.html.**