The Gentle Wake Up Alarm Clock
Explainer Video Script (2min 57sec):

My project is the gentle wake up alarm clock, a device designed to facilitate a smooth transition from sleep to wakefulness without inducing stress. Initially, I proposed a range of complex features intended to support this seamless transition; however, during the coding process, adjustments were made to accommodate a feasible timeline and the limitations of the available equipment.

Throughout the project, I used the Arduino programming environment to interface with various hardware components, including an Arduino Uno microcontroller, an LCD display, push-buttons, a real-time clock (RTC) module, a buzzer, LEDs, resistors, and connecting wires. To enable communication with these components, I installed and utilised several Arduino libraries, such as LiquidCrystal, Wire.h, and RTClib, each of which facilitated interaction with specific hardware elements.

The code was structured in a chronological setup sequence, initializing all components to operate once properly configured. The libraries simplified the tasks of timekeeping, controlling hardware, and displaying information on the LCD. To synchronize all components effectively, I followed a step-by-step tutorial available on YouTube, which guided the integration of multiple hardware elements using Arduino coding.

Over the course of the semester, I relied on both in-class resources and independent research to expand my understanding of C++ programming. Online resources and tutorials were particularly valuable in refining the final version of the code. For example, lines 1–42 of the program were dedicated to preparing essential elements of the alarm clock, including setting up the display, defining button functions, establishing melody notes, and configuring the alarm time.

In assembling the physical components, I began by connecting the Arduino Uno board to a power source via my computer, which also allowed connection to the Arduino application. Next, I set up the LCD display using its four wires—ground, 5V, SDA, and SCL—to provide power, grounding, and links for data and clock signals. Other components, including push-buttons, the RTC module, and the buzzer, were connected via a breadboard. Accurate placement of each component in the physical setup was essential, as their positions needed to correspond to the assignments in the Arduino code. Despite multiple attempts, I was unable to run the full alarm clock successfully; although the code verified correctly, I suspect that misalignment or incorrect positioning of hardware components caused the failure.

Although I was unable to synchronize all elements fully, I successfully demonstrated individual functionalities, such as playing music, lighting an LED, and initializing the LCD display.

With additional time for research and development, I believe I could have achieved full system integration. Nevertheless, the project provided valuable lessons in refining goals based on technical constraints, troubleshooting, and managing time effectively. While the final outcome differed from my original vision, the process reinforced the importance of iterative problem-solving and adapting project scope to realistic parameters.