```latex
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx}

\usepackage{geometry}

\usepackage{titlesec}

\usepackage{enumitem}

\usepackage{hyperref}

\geometry{margin=1in}


\titleformat{\section}{\large\bfseries}{\thesection.}{0.5em}{}

\setlist[itemize]{leftmargin=*}


\begin{document}


\begin{center}

   \Large \textbf{IoT Based Water Level Monitoring and Pump Control using ESP8266 NodeMCU} \\[0.5em]

   \normalsize

   \textbf{Microproject Report}\\[0.5em]

   \textbf{Submitted by:} Shreya Ann Jogi, Sreeparvathy P, Ruth Sara Sebastian \\[0.5em]

   Department of Electronics and Computer Engineering \\[0.5em]

   \textit{(Insert College Name Here)} \\[1em]

\end{center}


\hrule

\vspace{1em}


\section{Introduction and Objectives}

\textbf{Abstract:}
```

This project presents an IoT-based system that automatically monitors and controls the water level in a tank using an ultrasonic sensor and NodeMCU ESP8266 microcontroller. The system measures the water level in real-time and uploads data to the ThingSpeak cloud platform. It automatically turns the water pump ON or OFF based on predefined water levels, ensuring efficient water usage and preventing overflow or dry run. The IoT integration enables remote monitoring through the internet.

\textbf{Objectives:}

\begin{itemize}

\item To design an automated water level monitoring system using IoT.

\item To interface an ultrasonic sensor with NodeMCU ESP8266 for real-time level detection.

\item To control a water pump automatically through a relay module.

\item To send water level data to ThingSpeak for online visualization.

\item To minimize manual intervention and prevent water wastage.

\end{itemize}

\section{Block / Functional Diagram and Explanation}

\begin{center}

\includegraphics[width=0.8\linewidth]{block_diagram_placeholder.png}\\

\textit{(Figure 1: Functional Block Diagram of IoT-Based Water Level Monitoring System)}

\end{center}

\textbf{Explanation:}

\begin{itemize}

\item The \textbf{ultrasonic sensor (HC-SR04)} measures the distance between the sensor and the water surface.

\item The \textbf{NodeMCU ESP8266} processes this data and calculates the water level percentage.

\item When the water level drops below a set threshold, the NodeMCU activates the \textbf{relay module} to switch the pump ON.

\item When the tank is full, the relay turns the pump OFF automatically.

\item The measured level data is transmitted to the \textbf{ThingSpeak cloud} via Wi-Fi for online monitoring.

\end{itemize}

\section{Circuit Diagram, Components and Working Principle}

\begin{center}

\includegraphics[width=0.8\linewidth]{circuit_diagram_placeholder.png}\\

\textit{(Figure 2: Circuit Diagram of the Proposed System)}

\end{center}

\textbf{Components Used:}

\begin{itemize}

\item NodeMCU ESP8266 Microcontroller

\item Ultrasonic Sensor (HC-SR04)

\item 5V Relay Module

\item Water Pump

\item Jumper Wires, Breadboard, Power Supply

\end{itemize}

\textbf{Working Principle:}

The ultrasonic sensor continuously measures the distance to the water surface. The NodeMCU converts this distance into a percentage representing the tank's fill level.

When the distance indicates a low level (below 35 cm), the relay energizes and the pump turns ON. When the level reaches the upper limit (below 8 cm), the relay deactivates and the pump turns OFF.

The NodeMCU also connects to Wi-Fi and updates the ThingSpeak channel with the water level, distance, and pump status every 15 seconds.

```latex
\section{Description of Developed Code and Interfacing Logic}

The code initializes the NodeMCU pins for the ultrasonic sensor and relay. It reads the distance using the \texttt{pulseIn()} function and converts it to a water level percentage.

Two thresholds are defined for pump control to avoid frequent switching — this is called \textbf{hysteresis logic}.


\textbf{Code Working Explanation:}

\begin{itemize}

   \item Connects NodeMCU to Wi-Fi using SSID and password.

   \item Reads distance from HC-SR04 every 2 seconds.

   \item Converts distance into tank level percentage.

   \item Turns the pump ON when level is low (distance $\geq$ 35 cm).

   \item Turns the pump OFF when level is high (distance $\leq$ 8 cm).

   \item Sends the data (level, distance, pump status) to ThingSpeak every 15 seconds.

   \item Displays readings and pump state on the serial monitor.

\end{itemize}


\section{Image of Working Model}

\begin{center}

   \fbox{

   \begin{minipage}[c][6cm][c]{0.8\linewidth}

   \centering

   \vspace*{2cm}

   \textit{Insert Image of Working Model Here}

   \end{minipage}

   }\\[1em]

   \textit{(Figure 3: Prototype of IoT-Based Water Level Monitoring System)}

\end{center}
```

```latex
\section{Expected and Observed Results / Outcomes}

\textbf{Expected Results:}

\begin{itemize}

  \item The system should accurately detect the water level using the ultrasonic sensor.

  \item The pump should turn ON and OFF automatically based on preset levels.

  \item Real-time water level data should be uploaded to the ThingSpeak cloud.

\end{itemize}


\textbf{Observed Results:}

\begin{itemize}

  \item The ultrasonic sensor provided stable readings of tank level.

  \item The relay module successfully controlled the pump as per logic.

  \item ThingSpeak dashboard displayed water level, distance, and pump state accurately.

  \item The system operated automatically without manual intervention.

\end{itemize}


\section*{Conclusion}

The developed IoT-based water level monitoring system effectively automates the process of
water tank management. The integration of NodeMCU and ThingSpeak enables continuous
monitoring and control through the internet. This system can be extended for smart home
and industrial water management applications.


\end{document}
```