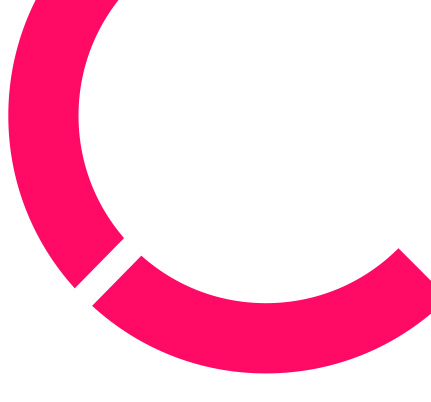
**Student’s name**

**ESSAY TITLE**

**Subheading if used**

**CENTRIA UNIVERSITY OF APPLIED SCIENCES**

**Month Year (e.g. September 2021)**

**CONTENTS**

[1 Abstract 1](#_Toc207017421)

[2 INTRODUCTION 2](#_Toc207017422)

[3 Project Overview 3](#_Toc207017423)

[4 Technologies Used 4](#_Toc207017424)

[4.1 Frontend 4](#_Toc207017425)

[4.2 Backend 4](#_Toc207017426)

[4.3 Database 4](#_Toc207017427)

[4.4 Hosting (Dev) 5](#_Toc207017428)

[5 Key Features 6](#_Toc207017429)

[5.1 Sensor Data Management 6](#_Toc207017430)

[5.2 Sensor Data Management 6](#_Toc207017431)

[5.3 Video Upload & Preview 6](#_Toc207017432)

[5.4 Storage Monitoring 6](#_Toc207017433)

[5.5 Storage Monitoring 7](#_Toc207017434)

[6 File Structure 8](#_Toc207017435)

[7 Testing & Validation 10](#_Toc207017436)

[8 UI Improvements 11](#_Toc207017437)

[9 Security Considerations 12](#_Toc207017438)

[9.1 Local Access Restriction 12](#_Toc207017439)

[9.2 Validation of File Type and Size 12](#_Toc207017440)

[9.3 Input Validation with FastAPI and Pydantic 12](#_Toc207017441)

[9.4 Directory Isolation for Uploaded Files 13](#_Toc207017442)

[9.5 Error Handling and Logging 13](#_Toc207017443)

[9.6 Export Function Restrictions 13](#_Toc207017444)

[9.7 Session Isolation 13](#_Toc207017445)

[9.8 Code-Level Security Practices 14](#_Toc207017446)

[10 Future Enhancements 15](#_Toc207017447)

[11 CONCLUSION 16](#_Toc207017448)

[REFERENCES 17](#_Toc207017449)

Using this template:

Since this template uses styles, the table of contents should not be edited manually, instead, it needs updating. Once the essay is ready, first left click the table of contents object so it gets highlighted, then right click on top, and choose “Update Field” from the menu, then click “Update entire table”.

This section above uses a line spacing of 1. The text in the rest of the template uses a line spacing of 1,5.

Based on this template you can start writing your essay, but remember to remove the unnecessary parts at the end.

# Abstract

This is a report of two standalone experimental studies designed to test simple concepts in physics. The first experiment is measuring Planck's constant, one of the building blocks of quantum mechanics, through measurements from a simulation or photoelectric effect experiment. Taking a series of readings of the stopping potential required to extinguish photoelectron emission as a function of incident wavelength of light, we calculated Planck's constant for each of the readings and then averaged the figure. The average Planck's constant we obtained was 5.05 x 10^-34 Js, which, while smaller than accepted, demonstrates the inherent proportionality of photon energy with wavelength as predicted by the photoelectric effect. This disparity can be attributed to explanations such as the omission of the work function in our calculations and experimental limitations.

The second experiment investigates the electrical characteristics of a Light Emitting Diode (LED), with emphasis on estimating its threshold voltage. Through the analysis of I-V data, we established the location where the LED begins to allow current to pass and emit light and estimated that the threshold voltage is approximately 2.5 V. Through this analysis, we are able to draw out the non-linear character of semiconductor devices as well as illustrate practical understanding of the operating characteristics of LEDs. Both experiments, as disparate as they are, help to inform a deeper understanding of quantum phenomena and semiconductor physics, reinforcing the necessity for empirical research in validating theoretical models.

# INTRODUCTION

This report comprises two independent experimental studies: the determination of Planck's constant based on the photoelectric effect and an estimate of the threshold voltage of an LED. Planck's constant, one of the most important quantum values, is explored based on the observation of the proportionality between photon energy and stopping potential and particle behavior of light. Concurrently, the electrical characteristic of an LED is explored in order to arrive at an estimate of its threshold voltage, one of the significant semiconductor parameters. These experiments bring together solid-state physics and quantum mechanics and offer valuable practical data regarding key physical principles. Through the integration of these studies, we are giving a full account of microscopic quantum phenomena and macroscopic electronic device behavior.

# Project Overview

The Admin Dashboard is a web-based interface designed to manage and visualize sensor data, video uploads, and storage usage for a data warehouse system. It provides real-time analytics, file management, and export capabilities for machine learning datasets

# Technologies Used

|  |  |
| --- | --- |
| Component | Technology |
| Frontend | HTML, CSS, JavaScript, Chart.js |
| |  |  | | --- | --- | | Backend |  | | FastAPI (Python) |
| Database | SQLite |
| Hosting (Dev) | Localhost via Uvicorn |

## Frontend

Explanation

## Backend

Explanation

## Database

Explanation

## Hosting (Dev)

Explanation

# Key Features

## Sensor Data Management

 Submit new sensor entries (ID, time range, value, unit)

 View sensor data in tabular format

 Delete old sensor data by timestamp

## Sensor Data Management

 Pie chart showing distribution of sensor types

 Line chart showing daily trends for selected sensors

## Video Upload & Preview

 Upload video files via form

 Preview videos inline using <video> tag

 Delete uploaded videos

## Storage Monitoring

 Live calculation of total used storage (in MB)

 Breakdown by category: Sensor DB, Video Files, Other Files

 Visualized using compact pie/bar charts

## Storage Monitoring

 Select sensor and time range

 Export data for machine learning training

# A screenshot of a computer AI-generated content may be incorrect.File Structure

A screen shot of a computer

AI-generated content may be incorrect.

# Testing & Validation

 Verified sensor data submission and deletion

 Confirmed video upload and preview functionality

 Validated storage breakdown accuracy via FastAPI logs

 Ensured charts render correctly and update live

 Tested public access from external devices via ngrok

# UI Improvements

 Reduced vertical spacing between charts and tables

 Aligned pie charts to the left for better layout

 Minimized chart height for compact dashboard

 Removed duplicate canvas elements

 Styled tables and forms for clarity and consistency

A screenshot of a computer

AI-generated content may be incorrect.

# Security Considerations

In the implementation of the Admin Dashboard for managing sensor data, video uploading, and machine learning data, multiple pragmatic security measures were incorporated to ensure secure and error-free functioning in a local environment. Despite the fact that the system is executed in a local server, the following security protocols were implemented to ensure data integrity, prevent misuse, and be prepared for any future public release.

## Local Access Restriction

The dashboard is locally hosted and accessible only within a safe network. This automatically limits the location vulnerable to harm. Steer clear of public hosting while in development also ensures that the possibility of unauthorized access, brute force, and data exposure is drastically reduced.

## Validation of File Type and Size

For preventing misuse of video uploading, validation was made very strict:

- Only `.mp4` are permitted.

- The size of the file is limited to a particular threshold so that the server is not overwhelmed by it or even malicious code.

This ensures that the system processes only relevant and safe content.

## Input Validation with FastAPI and Pydantic

All form inputs and API requests are validated using FastAPI’s built-in support for Pydantic models. This enforces strict data types and formats, reducing the risk of injection attacks or malformed data entries. For example, sensor data submissions are checked for correct structure before being stored or visualized.

## Directory Isolation for Uploaded Files

Uploaded video files are stored in a dedicated directory whose permissions for access are limited. This prevents the files from being run accidentally or exposed and ensures that the file system remains clean and secure.

## Error Handling and Logging

Basic error management mechanisms were implemented to prevent the system from crashing due to unexpected inputs or activities. Logs are generated for significant operations like file uploads, exporting data, and the generation of charts. These logs help track usage and identify potential misuse or bugs.

## Export Function Restrictions

The machine learning dataset export capability is limited to specific formats (e.g., CSV) and includes file protection against unauthorized access. Pre-approved datasets are allowed for download, and download activity is monitored to prevent excessive or bulk downloads.

## Session Isolation

User authentication is not present yet, but the dashboard is such that it will be able to run in isolated sessions. Every user action is stateless, and no sensitive information is cached between sessions. This minimizes the attack vector of session hijacking or exposure of data.

## Code-Level Security Practices

Secure coding practices were adhered to all over the codebase:

- Steering clear of hardcoded credentials or secrets.

- Configuration using environment variables.

- Keeping dependencies updated to patch known vulnerabilities.

These security controls provide a solid foundation for the existing use case of the Admin Dashboard. As the project progresses toward public release, more advanced controls such as user authentication, HTTPS encryption, and role-based access control will be included to meet more stringent security needs. The current implementation shows a fair balance between usability and safeguarding, keeping data secure and enabling proper analysis and management.

# Future Enhancements

 Deploy permanently to cloud (e.g., Vercel, Render)

 Add user authentication

 Enable chart export (PDF/PNG)

 Add historical storage tracking

 Integrate with external sensor APIs

# CONCLUSION

.

# REFERENCES

This section uses a line spacing of 1. The sources are listed according to the Harvard style, using a name-year system. More information can be found in the [Thesis Guidelines](https://libguides.centria.fi/centriaguideforthesis).

The sources are listed in alphabetical order. All sources listed must be cited in-text.

Leave one blank line between the sources.