TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

DHARAN

ID-CARD-GENERATOR

A COURSE PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS

AND COMPUTER ENGINEERING IN THE PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE PARTICAL COURSE ON

C++ PROGRAMMING LANGUAGE

Submitted By:

Anuj Kumar Thakur (PUR081BCT006)

Aaditya Kumar Karn (PUR081BCT001)

Bikash Bist (PUR081BCT012)

Submitted to:

Department of Electronics and Computer Engineering , Purwanchal Campus

Institute of Engineering, Tribhuvan University

Dharan, Nepal

Bhadra-03,2082

Acknowledgment

We would like to express our sincere gratitude to Mr. Tantra Nath Jha members for his invaluable guidance and support throughout our C++ programming language project. His insightful suggestions, constructive feedback, and unwavering encouragement were instrumental in our project. His expertise and mentorship have been truly inspiring, and we are grateful for his contribution to our learning journey.

Contents

Introduction

We propose to develop an interactive **ID Card Generator** using C++ with OpenCV. The primary goal of this project is to provide a simple yet efficient platform to generate personalized ID cards. Users can input their personal details, capture a photo using the webcam, and generate a visually appealing ID card automatically. The generated ID cards are saved with a timestamp for easy tracking.

Key Features:

- User-friendly interface to input personal details.
- Photo capture using a webcam.
- Automatic ID card generation with personalized details.
- File-based storage of ID cards and photos.
- Timestamped filenames for organization.
- Display of generated ID cards on the screen.

Objectives

1. To develop a fully functional ID card generator.
2. Practical application of C++ concepts:
   - File handling (imwrite, filesystem::create_directories).
   - User input handling (getline, cin).
   - Time and date manipulation (time.h, strftime).
3. To integrate OpenCV for image capture and processing.
4. To provide an easy and automated method to generate ID cards with photo and details.
5. To manage files and folders dynamically, ensuring proper organization.

Features of the Project

- **Photo Capture:** Using the webcam, users can capture their photo for the ID card.
- **Dynamic File Storage:** Each user gets a separate folder with timestamped files.
- **ID Card Layout:** Includes personal details like Name, ID, Position, Email, Phone, and Address.
- **Centered Text Placement:** Text automatically aligned to the center of the card.
- **Color-Coded Design:** Background and border colors for better aesthetics.
- **Timestamped Files:** Ensures unique filenames for photos and ID cards.
- **Display Functionality:** ID cards can be displayed directly using OpenCV.
- **Menu-Based Interaction:** Users can easily navigate through options like capturing a photo, generating an ID card, or displaying it.

System, Libraries, and Functions

The ID Card Generator is a console-based application using **C++ and OpenCV**. It is designed for Windows OS but can be adapted for Linux/macOS with minor changes.

Libraries Used:

- opencv2/opencv.hpp → Image capture, manipulation, and display.
- iostream → Standard I/O.
- string → String handling.
- ctime → Timestamp generation.
- sstream & iomanip → Formatting timestamps.
- filesystem → Creating directories and managing files.

Key Functions:

1. getTimestamp() → Returns current date and time as a string.
2. capturePhoto(Person &person) → Opens webcam, captures photo, saves in the user folder.
3. generateIDCard(const string &photoFile, Person &person) → Creates an ID card with photo and details.
4. showIDCard(const string &idCardFile) → Displays the ID card on screen.
5. getCenteredTextPosition() → Calculates text position to center-align on the card.
6. File handling functions: imwrite() (saving images), filesystem::create_directories() (folder management).

Methodology

Development Tools

- **Programming Language:** C++
- **IDE:** Visual Studio Code
- **Libraries:** OpenCV for image processing, standard C++ libraries for file handling and input/output.

Development Process

1. **Planning Structure**
   - Identify all required fields (Name, ID, Program , Roll no , Faculty , Email, Phone, Address).
   - Design card layout (size, photo placement, text spacing, colors).
   - Decide on menu-driven user interaction for capturing photos and generating ID cards.
2. **Setting Up Environment**
   - Install OpenCV and configure it with Visual Studio Code.
   - Set up necessary header files and libraries for compilation.
3. **Reading and Storing User Data**
   - Use a Person struct to hold all details.
   - Input handled via getline for proper handling of strings with spaces.
   - Folder created dynamically for each user to store photos and ID cards.

4. **Photo Capture**
    - Use OpenCV VideoCapture to access webcam.
    - Display live feed until user presses 'c' to capture.
    - Save photo with timestamped filename.

5. **ID Card Generation**
    - Create blank card image with background color.
    - Draw border and insert photo.
    - Place text using putText() with center alignment.
    - Save ID card image with timestamped filename.
6. **Display and Interaction**
    - Menu-based interface for navigation.
    - Users can choose to capture photo, generate ID card, or display card.
    - Input validation ensures smooth interaction.
7. **File Management**
    - Create separate folder per user.
    - Timestamped filenames prevent overwriting existing files.
    - Organize photos and ID cards efficiently.
8. **Testing**
    - Check photo capture and saving functionality.
    - Verify ID card generation with proper alignment and text formatting.
    - Ensure menu-driven system works correctly.
    - Test file and folder creation to handle multiple users.

PROJECT SCOPE

Objectives

- Main Goal: Develop a simple and interactive ID card generator for personal or organizational use.
- Target Audience: Students, employees, or anyone needing a digital ID card.

Key Features

- Menu-driven console interface.
- Personalized ID cards with photo and details.
- Automatic folder and file management.
- Timestamped filenames to avoid duplicates.
- Display functionality using OpenCV.

Exclusions

- No GUI beyond OpenCV image display.
- No network connectivity or cloud storage.
- No printing functionality (can be added later).
- No QR code or barcode integration (future enhancement).

PROJECT SCHEDULE

Table 1: d

Week	Tasks
Week 1	Requirement analysis, struct design, and OpenCV setup.
Week 2	Implement photo capture and folder management.
Week 3	ID card generation and text formatting.
Week 4	Implement menu system, testing, and bug fixing.
Week 5	Final testing, optimization, and documentation.