



MOHAMED SATHAK ENGINEERING COLLEGE

KILAKARAI-623 806, RAMANATHAPURAM DIST.

Approved by AICTE, COA, New Delhi, DGS Mumbai, Affiliated to Anna University,
An ISO 9001:2015 Institution, Sponsored by Mohamed Sathak Trust, Chennai - 06.



(Recognized under section 2(f) & 12B of UGC, NewDelhi)

DEPARTMENT OF INFORMATION TECHNOLOGY



PROJECT - BadgeGen

2024-2025

INDEX

S.no	Contents
1	Team Details
2	Project Details
3	Declaration form

Team Details

1. Guide:-

Samundeeswari K

Assistant Professor (B.Tech Information Technology)

Mohamed Sathak Engineering College

Kilakarai.

2. Students Details

Pranav Gautham M (final year of B.Tech Information Technology)

Deepika S.S (final year of B.Tech Information Technology)

Srikavi A (final year of B.Tech Information Technology)

Harini Priya M (final year of B.Tech Information Technology)

PROJECT - BadgeGen

What is BadgeGen :

BadgeGen is a combination of two words:

1. Badge: Refers to an identification card or label, typically used to represent a person's credentials, such as an employee ID, student ID, or access pass.
2. Gen: Short for "Generator" or "Generation," indicating that the project is focused on generating or creating something—in this case, ID badges.

So, BadgeGen essentially means "Badge Generator," highlighting the core function of your project: creating digital ID badges. It's a straightforward and modern name that conveys the purpose clearly and efficiently.

This project consist of 6 sections like

- **Introduction**
- **Objective**
- **Methodology**
- **Work Plan**
- **Budget**
- **Additional Details**

1. Introduction

The Virtual ID Card Generator is a Python-based application designed to streamline the creation of digital identification cards. This tool can be utilized by organizations such as businesses, educational institutions, or events that require managing and generating ID cards for their members, employees, or participants.

The system operates by storing user information, including names, departments, email addresses, phone numbers, and photo paths, in an SQLite database. Upon request, it retrieves this information, generates a barcode corresponding to the user's unique ID, and combines these elements into a professional-looking ID card image. The final output is a digital ID card that can be stored or printed.

With an emphasis on efficiency and automation, the Virtual ID Card Generator reduces the time and effort needed to create ID cards manually. It leverages Python's powerful libraries to handle image processing and barcode generation, providing a flexible and scalable solution for a variety of contexts.

2. Objectives

The main objectives of this project are:

- **Automate ID Card Creation:** Eliminate the need for manual design and creation of ID cards by automating the process through code.
- **Centralize User Data:** Store user data in a single, lightweight database (SQLite) for easy access and management.
- **Barcode Integration:** Integrate barcodes into the ID cards to provide a more secure and machine-readable method of identifying individuals.
- **User-Friendly Output:** Generate ID cards in a digital format that can be easily stored, shared, or printed.
- **Customization:** Allow for customization of the ID card's design, including the layout, fonts, colors, and images, to match an organization's branding.

3. Methodology

The methodology for the Virtual ID Card Generator involves several key steps:

1. Data Storage:

- **Database Setup:** An SQLite database named `id_cards.db` is used to store user information. The database includes a `users` table with fields for user ID, name, department, email, phone, and the path to their photo.
- **Data Entry:** User details are entered into the database either through the application or by directly inserting data using SQL commands.

2. **Data Retrieval:**

- **Fetch User Details:** When an ID card needs to be generated, the application queries the SQLite database to fetch the user's details based on their unique ID.
- **Handle Missing Data:** The system ensures that if user details are missing or the ID is not found, appropriate error messages are displayed.

3. **ID Card Generation:**

- **Image Creation:** Using the Pillow library (PIL), a blank ID card template is created with predefined dimensions and background color.
- **Photo Integration:** The user's photo is retrieved from the specified file path and resized to fit into the ID card layout.
- **Text Placement:** The user's details, such as name, ID, department, email, and phone number, are added to the card using text rendering functions in PIL. Font choices and sizes are defined for a clean and professional appearance.
- **Barcode Generation:** A barcode is generated based on the user's ID using the python-barcode library. The barcode is saved as an image and integrated into the ID card.
- **Saving the Card:** Once the card is fully assembled, it is saved as an image file (e.g., PNG), ready for use.

4. **Error Handling and Validation:**

- The system includes error handling mechanisms to ensure robustness. For example, it checks if the user ID exists in the database, handles file path issues for photos, and manages database connection errors.

5. **Output:**

- The final output is a digital ID card image that can be stored electronically, shared via email, or printed. The output format (e.g., PNG, JPEG) can be adjusted based on the requirements.

4. Work Plan

The work plan for the Virtual ID Card Generator is divided into four main phases:

1. **Phase 1 - Design and Setup:**

- **Database Design:** Develop the schema for the SQLite database to store user information.
- **Environment Setup:** Install Python and necessary libraries (PIL, SQLite, Barcode) on the development machine.
- **Initial Data Insertion:** Insert sample data into the database to be used for testing the ID card generation process.

2. **Phase 2 - Core Development:**

- **ID Card Layout Design:** Design the layout for the ID card, including

the placement of user details, photo, and barcode.

- **Barcode Integration:** Implement the functionality to generate and add a barcode to the ID card based on the user's unique ID.
- **User Photo Integration:** Develop code to retrieve and resize the user's photo for placement on the ID card.

3. **Phase 3 - Testing and Optimization:**

- **Testing with Sample Data:** Test the application with different sets of sample data to ensure all components (photo, text, barcode) are correctly integrated into the ID card.
- **Error Handling:** Test for potential errors, such as missing user data or incorrect file paths, and implement error handling to manage these cases.
- **Performance Optimization:** Optimize the code to improve performance, especially when generating multiple ID cards in batch mode.

4. **Phase 4 - Finalization and Documentation:**

- **Documentation:** Create detailed documentation explaining how the system works, including setup instructions, code explanations, and usage guidelines.
- **Final Testing:** Conduct a final round of testing to ensure all features are functioning as expected and the system is ready for deployment.

5. Budget

This project relies on open-source tools and libraries, keeping costs minimal. However, some budget considerations include:

- **Development Costs:** The time spent by the developer(s) to write, test, and document the code. Potential costs associated with hiring additional developers or consultants for specialized tasks.
- **Hardware Costs:** If running the project on a local machine, the hardware should be capable of handling image processing tasks. This may require investment in a moderately powerful computer with sufficient memory and storage.
- **Software Costs:** No direct software costs are involved, as the project uses free, open-source libraries such as Pillow for image manipulation, SQLite for database management, and python-barcode for barcode generation.

6. Additional Details

- **Scalability:** The system can be expanded to accommodate a larger user base by scaling the database and optimizing code performance. Additional features, such as batch processing for multiple ID cards, can also be integrated.
- **Customization:** The ID card layout can be tailored to meet specific

organizational needs, such as different font styles, colors, or additional fields (e.g., job title, location). The system can also be expanded to support multiple ID card templates for different departments or user types.

- **Security:** Enhancing security by incorporating encrypted QR codes, holograms, or watermarks is a potential future improvement. Additionally, user data can be protected by implementing encryption in the database and secure access controls.
- **Future Enhancements:**
 - **Web Interface:** A web-based front-end could be developed to allow users to generate ID cards through a browser interface, making the system accessible without needing to interact directly with the code.
 - **Mobile Compatibility:** Creating mobile-friendly ID cards or even digital IDs that can be stored in a mobile app wallet (e.g., Apple Wallet, Google Pay).
 - **Integration with External Systems:** The system could be integrated with external HR or student management systems to automatically update and generate ID cards when new users are added.

Declaration form

