Software Design Documentation

Project Name: Alumni Tracker
Date: 2080/05/11

Project Contributors:
Nayan Pandey (077bct049)
Nirmal Rana (077bct051)
Prasun Sitaula (077bct057)
Rajesh Adhikari (077bct065)

Introduction:

This project is an Alumni Tracker that aims to simplify the process of tracking and managing alumni records for the Department of Electronics and Computer Engineering (DOECE) at Pulchowk Campus, IOE. With a history of offering programs since 1994, the department recognized the need for an organised system to maintain alumni data. The project's primary objective is to develop a web-based system that integrates alumni information into a well-structured database and provides a portal for alumni to update their current status and access online yearbooks.The project was initially created by Anusandhan Pokhrel, Baibhav Bista, Lumanti Dangol, and Mohit Kedia. The project is deployed in the CIT server using plithos web hosting service. The url of the project is DOECE Alumni Portal (itclub.pp.ua).

Features:

Alumni Records Management

The system allows for the seamless integration of alumni records from various sources. It includes data collection, cleaning, and integration processes to maintain accurate and consistent records.

Online Yearbook Publication

The Alumni Tracker generates online yearbooks that showcase alumni profiles, achievements, and updates. These yearbooks serve as a valuable platform for alumni to reconnect and reminisce.

Self-Update by Alumni

Alumni can log in to the system and update their personal and professional information. This feature ensures that the database remains current and provides incentives for alumni to stay engaged.

Admin Panel
Admin users have access to a dedicated admin panel that offers advanced search, filtering, data import/export, and database management functionalities. This panel empowers admins to maintain the database efficiently.

Additional Features

1. Signup and login feature for students using email/password credentials
2. Custom Django app for viewing and editing student data
3. ERD of Database Models
4. Filter system on the basis of name,country, employed organisation and university.

Technologies Used:
Languages: Python
Frameworks: Django
Database: sqlite3
CSS library: Bootstrap 4

Installation:

- Create a user group 'Students' and a group 'Institutes' in the Django admin panel.
- Create a user and add them to the 'Institutes' group.
- Create an entry in the 'Institutes' model using the above user.
- Access institute facilities using the URL <host>/institute.
- Main Django app: DOECEAlumniStudent
- Records app: records
- Make sure to have included all the installed apps in settings.py.

```
INSTALLED_APPS = [
    'records',
    'institutuff',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.sites',
    'allauth',
    'allauth.account',
    'allauth.socialaccount',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'advanced_filters',
    'django_filters',
    'crispy_forms',
    'dbbackup',
]
```

- Configure the database settings in DOECEAlumniStudent/settings.py.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

- Put the media folder in the root directory. Create another directory inside media called documents_photos.
- For security, ensure database credentials are stored as environment variables.
- Do not include venv and media directories in the repository(unless necessary).
- Make sure the allowed host is set so that everyone can visit the site.

```
ALLOWED_HOSTS = [
    '127.0.0.1',
    '*'
]
```

Tutorial Video:
(https://drive.google.com/file/d/1JpWRs7erkSG8iWnwYFkEz-mo-UhmfN5U/view?usp=sharing)

Configuration:
Before running the project, make the following configurations:

- Create user groups 'Students' and 'Institutes' in the Django admin panel.
- Create a user and add them to the 'Institutes' group.
- Create an entry in the 'Institutes' model, assigning the user from the previous step.
- Configure database settings in DOECEAlumniStudent/settings.py.
- If you want to use POSTGres as a database, follow the above tutorial video for database configuration.

Running the Project:

- Clone the repository https://github.com/PRASUN-SITAULA/Alumni-Tracker.git
- Method 1:
  - Create a virtual environment and install the packages from requirements.txt using command *pip install -r requirements.txt.*
  - If the project gives a package installation error, upgrade to the latest version of python.
  - Run *python manage.py makemigrations* followed by *python manage.py migrate*
  - Finally run *python manage.py runserver.*
  - Visit http://localhost:8000
- Method 2:
  - Install Docker on your system.
  - Run the docker engine.
  - Navigate to the root directory containing the docker-compose.yml file.
  - Run the command *docker compose up*
  - Visit the url http://localhost:7071 or http://127.0.0.1:7071

System Overview:

Frontend

The Alumni Tracker features a publicly hosted frontend user interface accessible via web browser. This interface provides access to online yearbooks for public users and allows alumni to login and update their information. The frontend is built using modern web technologies to ensure a user-friendly and responsive experience.

Backend

The backend of the Alumni Tracker handles incoming requests from the frontend, processes application logic, and interfaces with the database. It ensures seamless communication between user actions and the database, facilitating operations such as updating alumni information, generating yearbooks, and performing advanced searches.

Database Structure

The heart of the Alumni Tracker is the database, where all alumni records are stored. The database stores various attributes such as personal information, contact details, academic history, employment status, and more. The schema is designed to accommodate different types of data while ensuring data integrity and consistency.

The database has a simple design, with a main ‘records student’ table and separate tables for attributes which may have multiple instances – address, social media, enrolled programs (in DOECE) and further academics, stored in tables ‘records addresses’, ‘records socialmedia’, ‘records enrolledprograms’ and ‘records furtheracademicstatus’ respectively. The project's database structure is visually represented in the ERD diagram. It includes tables for student data, user groups, and more.

The system is designed for use by three categories of users: Public, Alumni and Admins. Public users can view yearbooks. Alumni users can login and update their information in addition to being able to view yearbooks as mentioned before. Admin users have the most privileges and have access to admin dashboard with features like advanced search and functionality like import and export data.

System Architecture:

The system architecture of the Alumni Tracker is designed to ensure efficient communication, data storage, and user interaction. It consists of three core components: a frontend, backend, and database as mentioned before. The frontend provides an intuitive user interface accessible via web browsers, allowing alumni to interact with the system. The backend handles the processing of user requests, logic implementation, and data manipulation, serving as the bridge between the frontend and the database. The database is the central repository where all alumni records are stored, organised, and managed. This architectural structure guarantees seamless user experiences, data integrity, and effective management of alumni information.

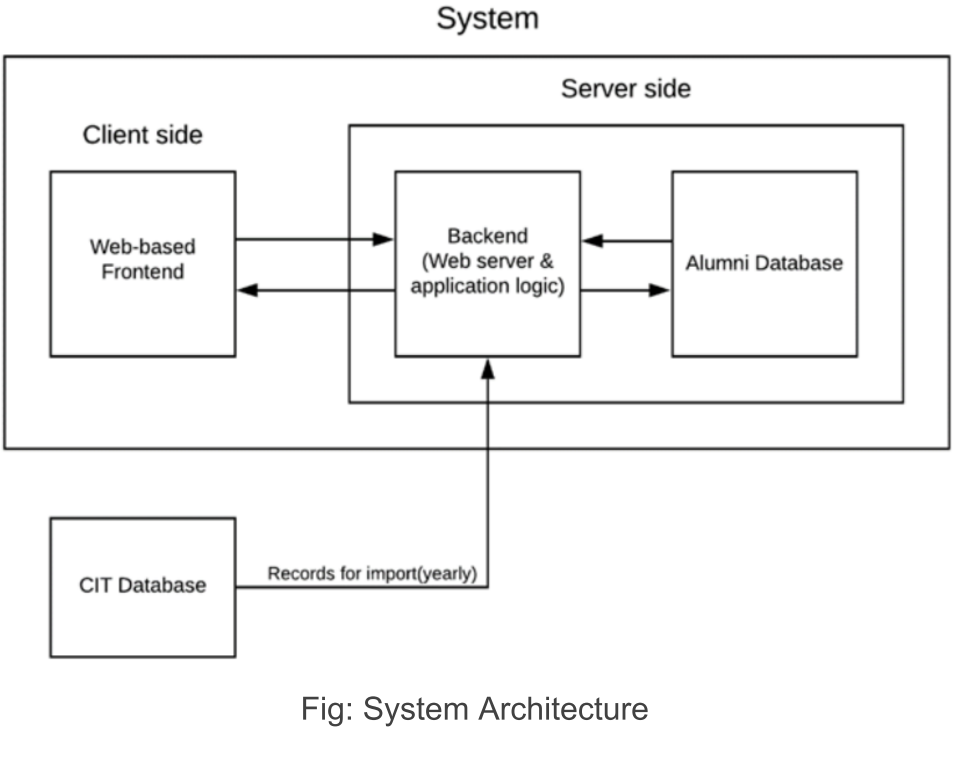


Fig: System Architecture

Use Case Diagram:

The use case diagram illustrates the interactions between different user roles in the Alumni Tracker system: public users, alumni, and admins. It depicts actions such as viewing yearbooks, updating information, and managing the database.

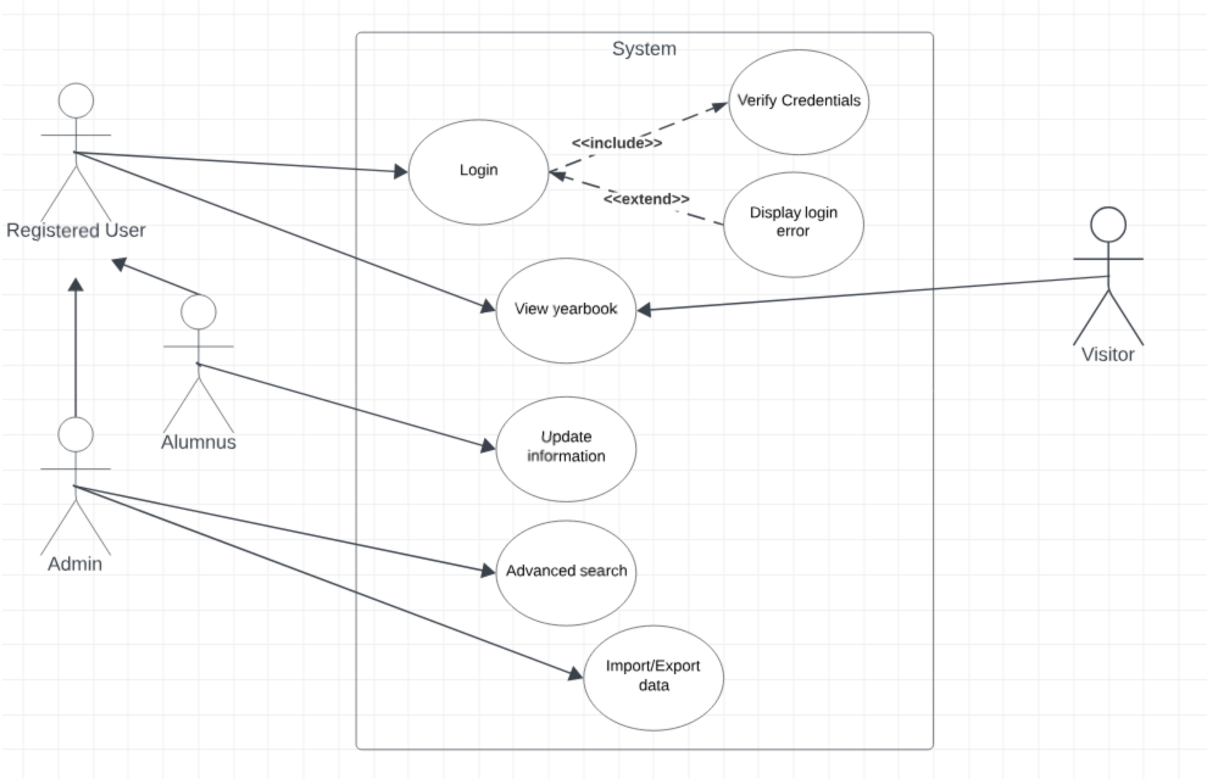


Fig: Use Case Diagram

Conceptual Database Design:

The database design is structured to store alumni records efficiently and accurately. It includes tables for personal details, contact information, academic history, employment status, and further education. The schema is designed to support relationships between different attributes and entities.

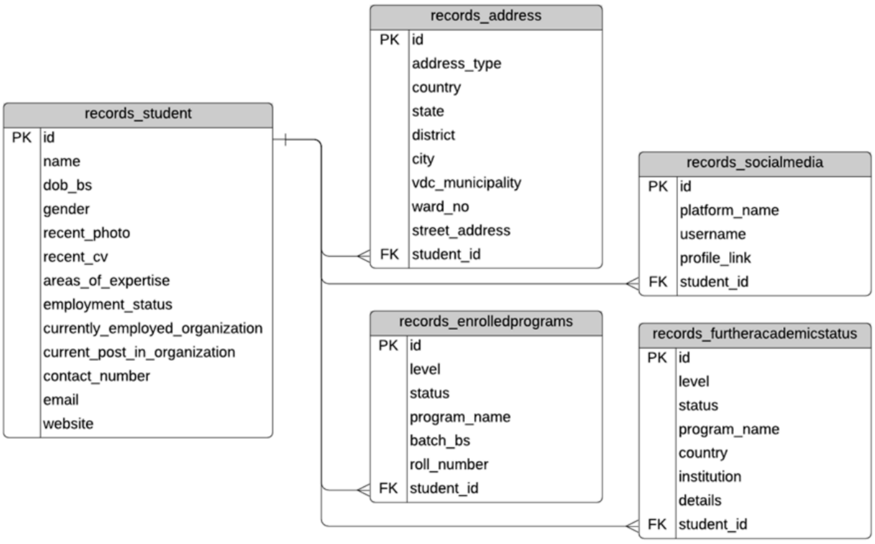
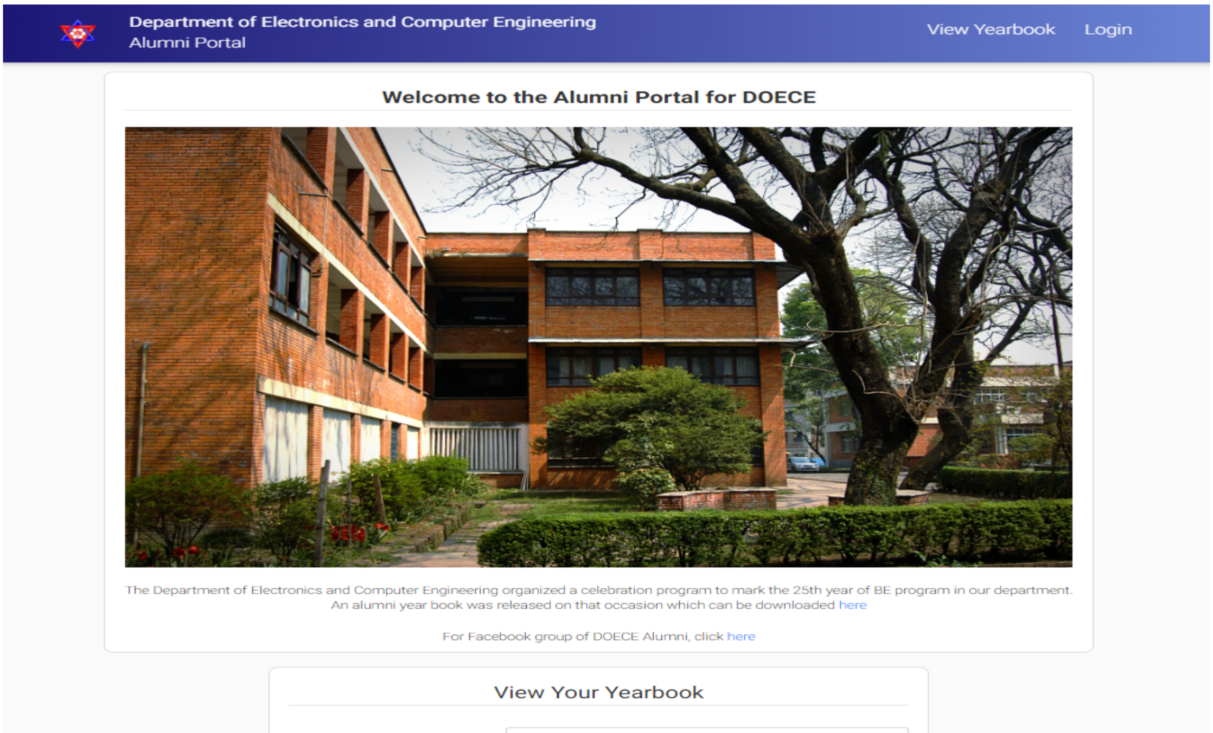


Fig: Schema Design

Usage:

Navigate to the project homepage to explore its features and functionalities, and utilise the advanced search system to effortlessly locate alumni based on various criteria, login to update the information:

Homepage:



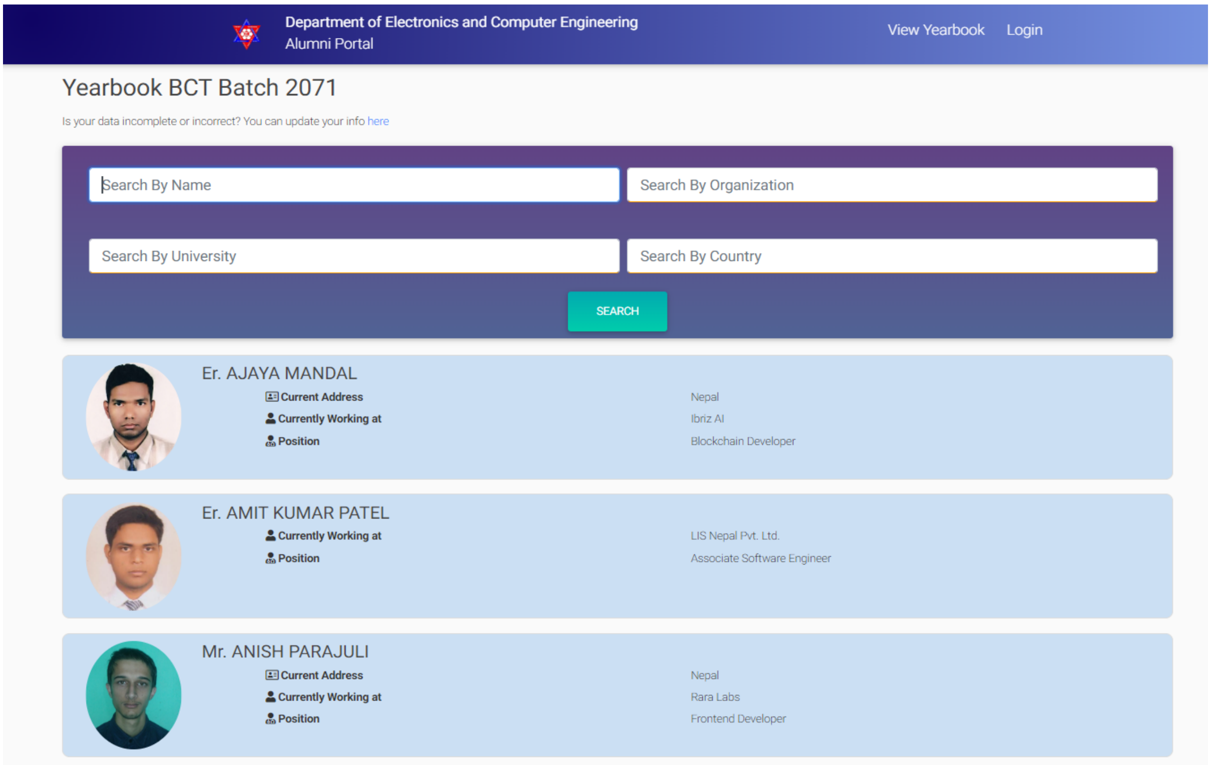
Welcome to the Alumni Portal for DOECE

The Department of Electronics and Computer Engineering organized a celebration program to mark the 25th year of BE program in our department. An alumni year book was released on that occasion which can be downloaded here

For Facebook group of DOECE Alumni, click here

View Your Yearbook

Search System:



Yearbook BCT Batch 2071

Is your data incomplete or incorrect? You can update your info here

Er. AJAYA MANDAL

Current Address	Nepal
Currently Working at	Ibriz AI
Position	Blockchain Developer

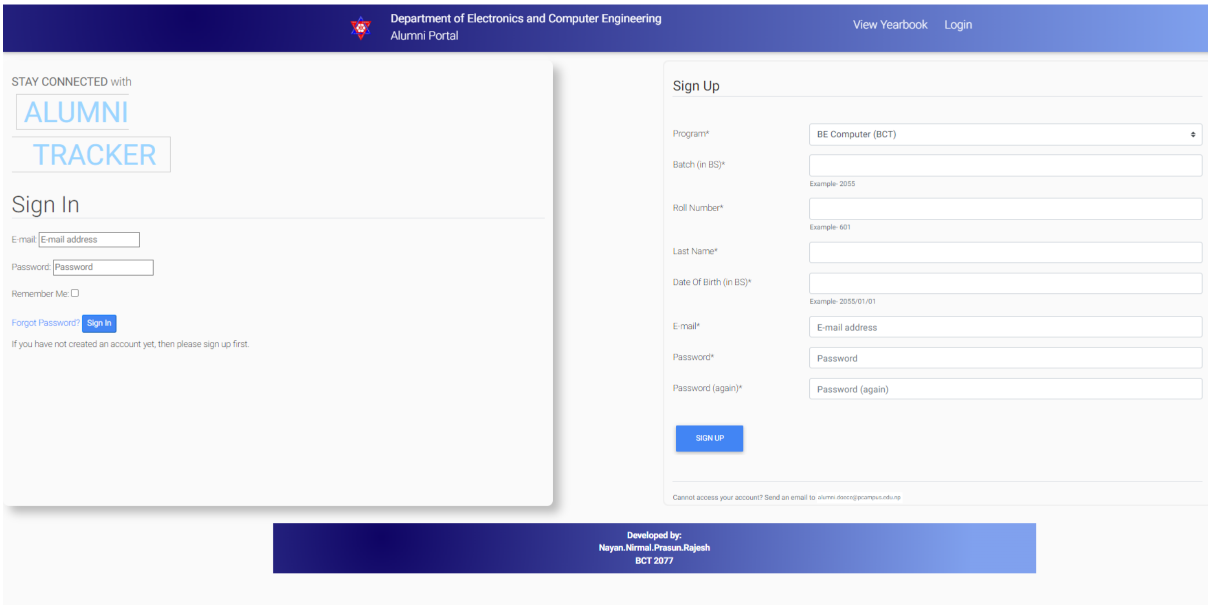
Er. AMIT KUMAR PATEL

Currently Working at	LIS Nepal Pvt. Ltd.
Position	Associate Software Engineer

Mr. ANISH PARAJULI

Current Address	Nepal
Currently Working at	Rara Labs
Position	Frontend Developer

Login:



STAY CONNECTED with

ALUMNI

TRACKER

Sign In

E-mail: E-mail address

Password: Password

Remember Me: □

If you have not created an account yet, then please sign up first.

Sign Up

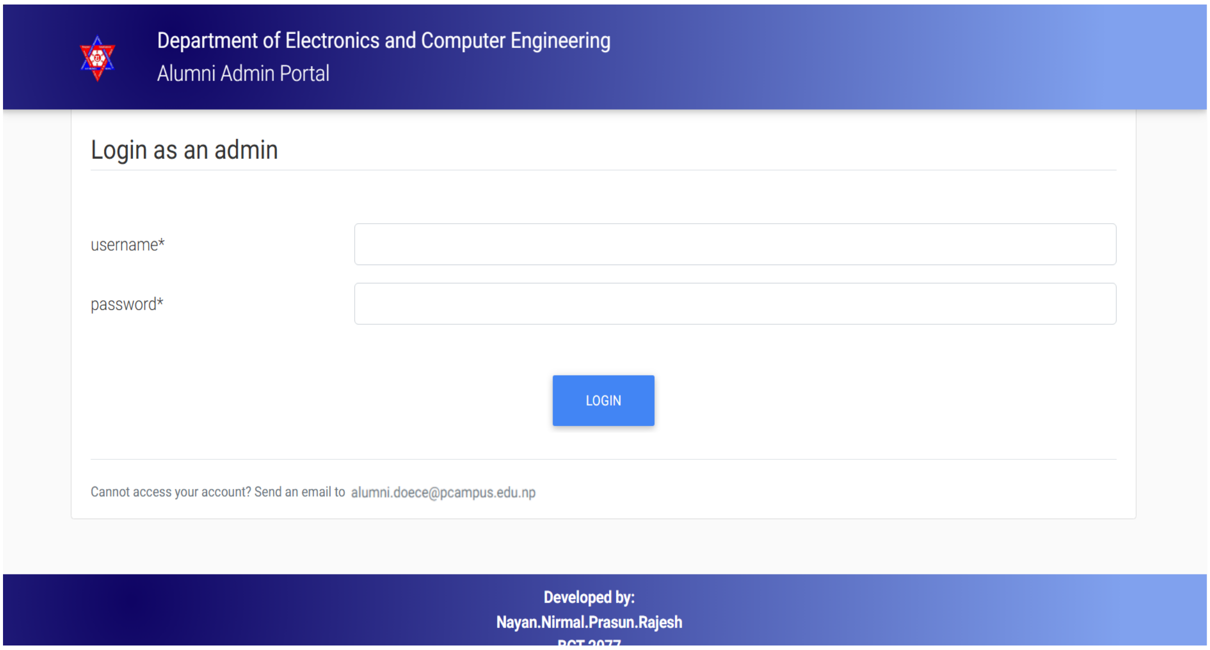
Program*	BE Computer (BCT)
Batch (in BS)*	Example- 2055
Roll Number*	Example- 601
Last Name*	
Date Of Birth (in BS)*	Example- 2055/01/01
E-mail*	E-mail address
Password*	Password
Password (again)*	Password (again)

Cannot access your account? Send an email to alumni.doece@pcampus.edu.np

Developed by:
Nayan.Nirmal.Prasun.Rajesh
BCT 2077

Authentication and Authorization:

The project employs authentication and authorization mechanisms. Students and authorised institute users have distinct access levels.



Department of Electronics and Computer Engineering
Alumni Admin Portal

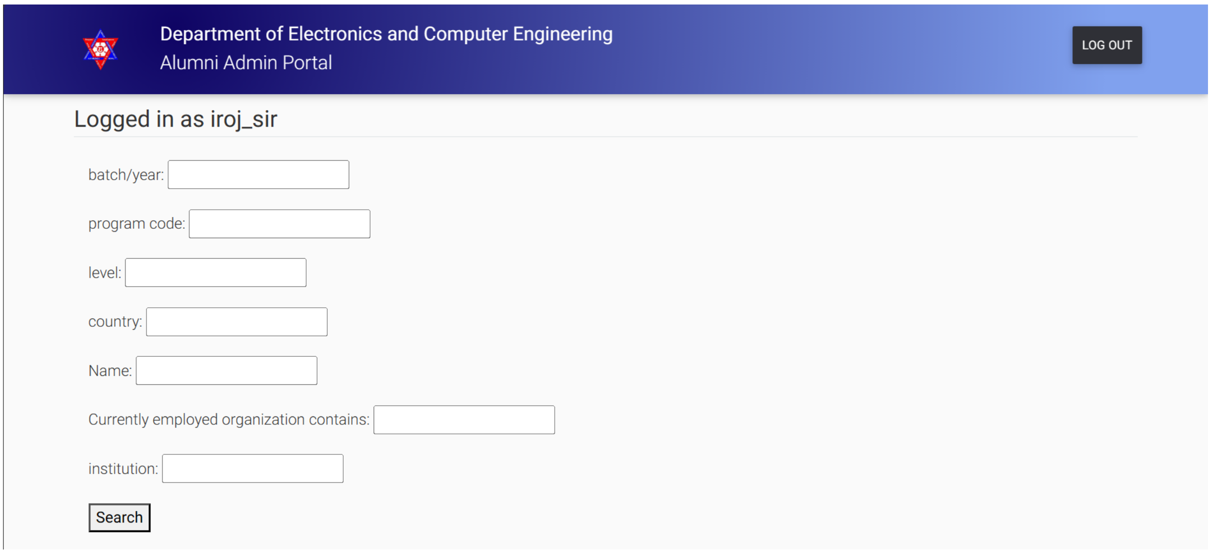
Login as an admin

username*

password*

Cannot access your account? Send an email to alumni.doece@pcampus.edu.np

Developed by:
Nayan.Nirmal.Prasun.Rajesh
BCT 2077



Department of Electronics and Computer Engineering
Alumni Admin Portal

Logged in as iroj_sir

batch/year:

program code:

level:

country:

Name:

Currently employed organization contains:

institution:

Deployment:

For deployment, follow these steps:

- Set up the production environment by configuring settings in settings.py.
- Create a DockerFile for containerization.
- Deploy the project on a server using the provided deployment tools.

Security Considerations:
To ensure the security of your application and data, follow these guidelines:

- **User Authentication**
  Implement strong user authentication to ensure only authorised individuals can access the system. Require secure passwords and consider implementing multi-factor authentication (MFA) for added security.

- **Data Protection**
  Encrypt sensitive data during transmission using HTTPS and store sensitive information, like passwords, using secure encryption techniques to prevent unauthorised access.

- **Input Validation**
  Validate and sanitise user inputs to prevent common security threats like hacking attempts and data breaches. This helps keep the application safe from attacks like SQL injection and cross-site scripting.

- **Sensitive Information Handling**
  Store sensitive information like database credentials and API keys as environment variables in the .env file, away from the source code, to prevent accidental exposure and unauthorised access.

- **Regular Backups**
  Regularly back up the database using the provided backup_script.sh to ensure data preservation and recovery in case of any unexpected data loss or security breach.

Contributing:
Contributions to the project are welcome. Potential contributions include:

- **Code Contributions:** If you're a developer, you can contribute to the project by submitting code improvements, bug fixes, or new features. Fork the repository, create a new branch for your changes, and submit a pull request. Make sure your code follows the project's coding standards and guidelines.
- **Bug Reports:** If you encounter any bugs, glitches, or unexpected behaviour while using the Alumni Tracker, please report them by opening an issue on the GitHub repository. Provide detailed information about the issue, steps to reproduce it, and any relevant logs or error messages.

- **Feature Requests:** Have an idea for a new feature or enhancement? Feel free to share it by opening an issue on the repository. Describe the feature request thoroughly, explaining its purpose and how it can benefit the project.

- **Documentation Improvements:** Clear and comprehensive documentation is crucial for any project. If you find areas where the documentation can be improved, or if you want to add new documentation, your contributions are highly appreciated.

- **Testing and Feedback:** Testing the application and providing feedback can help identify and address issues before they become problems for other users. You can help by testing new features, reporting any issues you find, and sharing your feedback.

Results and Analysis:

- **Alumni Data Insights**
  The Alumni Tracker system has collected a significant number of alumni responses. Preliminary insights include the distribution of alumni across various job positions, employers, and geographic locations. These insights provide valuable information for understanding alumni trajectories and preferences.

- **Visualisation of Insights**
  The system presents insights through visualisations such as bar charts and diagrams. These visualisations help in understanding trends related to alumni employment, residence, further education, and more.

- **Yearbook Creation**
  The Alumni Tracker system has been used to create online yearbooks that showcase alumni profiles and achievements. This feature has facilitated the celebration of the department's milestones and achievements.

Future Enhancements:
The Alumni Tracker project is continuously evolving, and there are numerous exciting possibilities for enhancing its capabilities.

- **Enhanced User Authentication Features:**
  One area of potential improvement is enhancing the user authentication mechanisms. This could involve implementing more advanced security measures, such as two-factor authentication, to ensure the privacy and integrity of user data. By bolstering authentication, we can provide alumni with even more confidence in the security of their information.

- **Advanced Data Visualization Tools:**
  Data visualisation plays a crucial role in conveying insights effectively. To this end, future versions of the Alumni Tracker could incorporate more advanced data visualisation tools. This might include interactive charts, graphs, and visual representations that allow users to explore alumni trends and statistics in a more intuitive and engaging manner.

- **Comprehensive Surveys and Feedback Mechanisms:**
  Expanding the system's capabilities for conducting comprehensive surveys among alumni could provide valuable insights into their preferences, needs, and expectations. By facilitating the collection of structured feedback, we can adapt the system to better align with the evolving requirements of alumni, making it an even more indispensable platform for communication and engagement.

- **Campus-wide Expansion:**
  While the current scope of the Alumni Tracker focuses on a specific department, there is potential to expand its reach campus-wide. By collaborating with other departments or institutions within the university, we could create a unified platform that serves as a central hub for alumni information, fostering broader connections and interactions.

- **Automated Data Collection from External Platforms:**
  Leveraging external platforms such as LinkedIn for automated data collection could be a game-changer. By integrating with these platforms, the Alumni Tracker could automatically update alumni profiles based on their professional information available on these networks. This would reduce the burden on alumni to manually update their details and ensure that the system remains up-to-date with accurate information.

Contact Information:
077bct049.nayan@pcampus.edu.np
077bct051.nirmal@pcampus.edu.np
077bct057.prasun@pcampus.edu.np
077bct065.rajesh@pcampus.edu.np