

## CAMPUS BUDDY

Sk. Shheema Zamell<sup>1</sup>, B. Mahesh<sup>2</sup>, B. Niharika<sup>3</sup>, B. Sri Harsha<sup>4</sup>, B. Akshay Kumar<sup>5</sup>,  
Dr. Gifta Jerith<sup>6</sup>

<sup>1,2,3,4,5</sup>B. Tech School of Engineering Computer Science – AI&ML Malla Reddy University, India.

<sup>6</sup>Guide: HOD School of Engineering Computer Science – AI&ML Malla Reddy University, India.

### ABSTRACT

CampusBuddy is an AI-powered application designed to enhance communication between students and professors in higher education institutions. Unlike traditional college websites, which often provide one-way communication, CampusBuddy enables dynamic two-way interaction, fostering collaboration. The app features an AI chatbot that offers instant assistance for student inquiries and navigation tools to help users find essential campus locations. Additionally, CampusBuddy provides timely updates on university events and important announcements, keeping students informed and engaged.

By fostering a sense of community and enhancing engagement, CampusBuddy not only supports academic success but also encourages student involvement in campus activities. The application's user-friendly interface and integrated features empower students to take an active role in their educational journeys. As educational institutions increasingly embrace technology, CampusBuddy stands out as a comprehensive solution that meets the evolving needs of modern learners, paving the way for a more connected and supportive educational environment.

**Keywords:** Python, Nlp

## 1. INTRODUCTION

In today's rapidly evolving educational landscape, effective communication between students and faculty is crucial for fostering academic success and engagement. Traditional college websites often fall short, providing only one-way communication that limits interaction and feedback. Recognizing this gap, CampusBuddy was developed as a comprehensive solution to enhance the educational experience through innovative technology.

CampusBuddy is an AI-powered application specifically designed for higher education institutions, aiming to facilitate seamless communication and interaction between students and professors. By integrating a two-way communication system, the app allows both parties to engage in meaningful dialogue, share feedback, and collaborate more effectively. This feature not only enhances the learning experience but also promotes a sense of community within the campus environment. In addition to its communication capabilities, CampusBuddy incorporates an AI chatbot that offers instant support for student inquiries, ensuring that help is always available for academic and administrative questions. The app also includes navigation tools to assist students in locating essential facilities on campus, such as classrooms and libraries, thereby easing the transition for new students and enhancing overall campus accessibility.

**Customer experience improvement:** To enhance the customer experience of CampusBuddy, we will focus on a user-friendly interface, personalized content, and a responsive AI chatbot for 24/7 support. Seamless communication between students and faculty will be prioritized, along with accessible onboarding tutorials and a feedback system for continuous improvement. Engaging community features and regular updates will ensure a smooth and efficient app experience, driving user satisfaction and engagement.

## 2. REQUIRED TOOLS

### a) Software Requirements

#### System Requirements

- **Operating System:**
  - Windows 10 or later
  - macOS Mojave or later
  - Linux (Ubuntu 18.04 or later)
- **Processor:**
  - Minimum: Dual-core processor
  - Recommended: Quad-core processor or higher
- **RAM:**
  - Minimum: 4 GB
  - Recommended: 8 GB or more

- **Storage:**

- Minimum: 100 MB of free disk space
- Recommended: 500 MB of free disk space for additional resources and data

## 2. Development Environment

- **Programming Languages:**

- Python 3.x for backend development
- JavaScript (Node.js) for server-side scripting
- HTML5, CSS3 for frontend development

- **Frameworks and Libraries:**

- **Backend:**

- Flask or Django (for web framework)
- SQLAlchemy (for database interactions)

- **Frontend:**

- React.js or Vue.js (for building user interfaces)
- Bootstrap or Tailwind CSS (for responsive design)

- **Database:**

- PostgreSQL or MySQL (for relational database management)
- MongoDB (for NoSQL data storage)

- **Data Analysis:**

- Pandas, NumPy, Scikit-learn for data manipulation and machine learning
- NLTK for natural language processing

## 3. Additional Software Tools

- **Version Control:**

- Git (for version control and collaboration)
- GitHub or GitLab (for repository hosting)

- **Integrated Development Environment (IDE):**

- Visual Studio Code, PyCharm, or any suitable IDE for coding

- **API Development and Testing Tools:**

- Postman or Insomnia (for API testing)

- **Containerization:**

- Docker (for creating, deploying, and managing containers)

## 4. User Requirements

- **User Interface:**

- A responsive design suitable for mobile and desktop platforms
- Intuitive navigation and accessibility features

- **Accessibility:**

- Compliance with accessibility standards (e.g., WCAG 2.1) to support users with disabilities

- **Security:**

- Implementation of security protocols to protect user data and privacy, including encryption and secure authentication methods.

### b) Hardware Requirements

#### Client-Side Hardware Requirements

- **Devices:**

- Smartphones (iOS and Android)
- Tablets
- Laptops and Desktops

- **Minimum Specifications:**

○ **Smartphones/Tablets:**

- Processor: Minimum quad-core
- RAM: 2 GB
- Storage: Minimum 16 GB of available storage

○ **Laptops/Desktops:**

- Processor: Dual-core processor (Intel i3 or equivalent)
- RAM: Minimum 4 GB (8 GB recommended)
- Storage: Minimum 100 MB of free disk space (SSD preferred for faster access)

● **Network Requirements:**

- Reliable internet connection (Wi-Fi or mobile data) with a minimum speed of 5 Mbps for optimal performance.

## 2. Server-Side Hardware Requirements

● **Web Server:**

- Processor: Minimum quad-core processor
- RAM: 8 GB (16 GB recommended for high traffic)
- Storage: SSD with a minimum of 500 GB for faster data access and scalability

● **Database Server:**

- Processor: Minimum quad-core processor
- RAM: 16 GB (32 GB recommended for larger datasets)
- Storage: High-performance SSD with 1 TB capacity for database management

● **Backup and Redundancy:**

- Separate backup server with equivalent hardware specifications to ensure data redundancy and recovery in case of failures.

## 3. Development Environment Hardware Requirements

● **Development Machines:**

- Processor: Quad-core or better (Intel i5 or equivalent)
- RAM: Minimum 8 GB (16 GB recommended for multitasking)
- Storage: SSD with at least 256 GB to accommodate development tools and project files

## 3. MODULES

**a) Natural Language Processing (NLP) Libraries:** CampusBuddy will utilize several Natural Language Processing (NLP) libraries to enhance its functionality. NLTK provides comprehensive text processing capabilities, while spaCy offers efficient named entity recognition and dependency parsing. For sentiment analysis, TextBlob serves as a user-friendly option, and Gensim specializes in topic modeling with algorithms like Word2Vec.

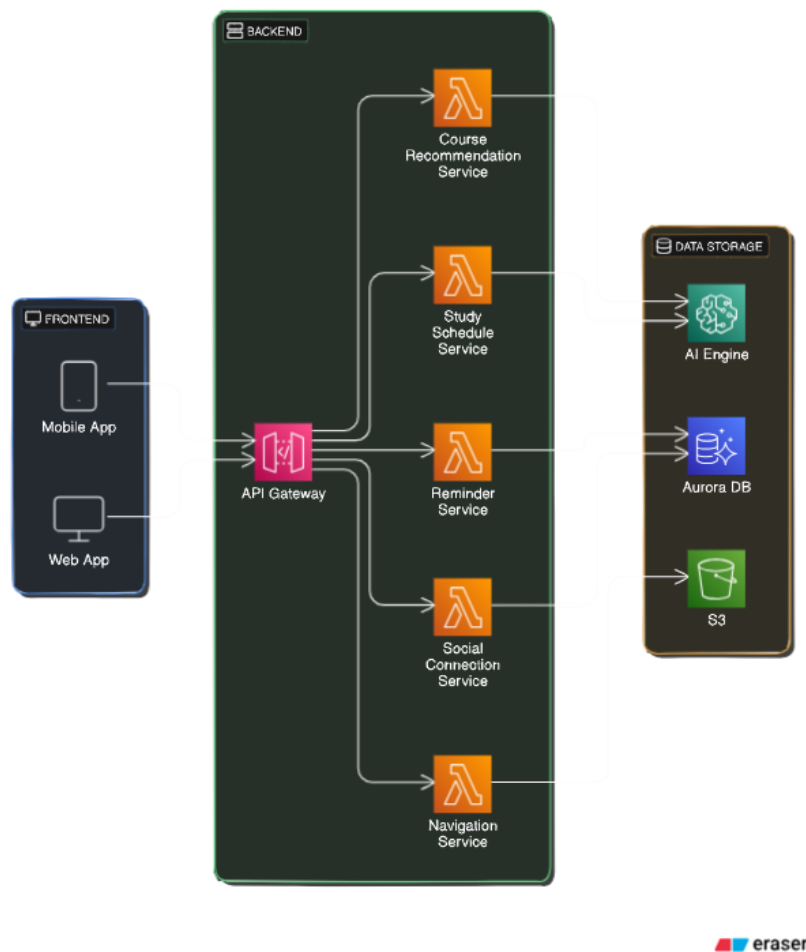
**b) Machine Learning Libraries-** CampusBuddy will utilize several machine learning libraries to enhance its functionality. Scikit-learn offers tools for classical algorithms and data preprocessing, while TensorFlow and Keras simplify deep learning model development. PyTorch provides flexibility with dynamic computation graphs for model construction. For boosting performance, XGBoost and LightGBM excel in speed and accuracy. Additionally, CatBoost effectively handles categorical features, enabling predictive analytics and personalized user experiences..

### Data Processing and Analysis:

- Pandas: A powerful library for data manipulation and analysis, often used for preprocessing and organizing data for sentiment analysis and user behavior insights.
- NumPy: A fundamental library for numerical computing, useful for handling arrays and performing mathematical operations, facilitating efficient data processing.

## ARCHITECTURE

CampusBuddy features a modular architecture with a responsive frontend built using React.js or Vue.js for seamless web and mobile interactions. The backend is powered by Flask or Django to manage business logic, API requests, and user authentication. Data is stored in PostgreSQL or MySQL for structured information, complemented by MongoDB for unstructured data. It integrates NLP libraries like NLTK and spaCy for processing user feedback and powering the AI chatbot.



Here are the main components of the architecture of a sentiment analysis:

- Data Collection:** Gather textual data from sources such as user feedback, academic forums, and communication logs. Use APIs or web scraping techniques to collect this data effectively.
- Feature Extraction:** Convert the preprocessed text into numerical features using techniques like TF-IDF or word embeddings to facilitate further analysis.
- Sentiment Analysis Model:** Train or utilize a pre-trained sentiment analysis model (e.g., using libraries like NLTK or spaCy) to predict the sentiment of user-generated text.
- Sentiment Classification:** Apply the trained model to feature vectors to classify sentiments as binary (positive/negative) or multi-class (e.g., happy, sad, frustrated) based on the context of the feedback.
- Post-processing and Analysis:** Evaluate and interpret the results of the sentiment analysis, calculating sentiment scores or confidence levels for each classified sentiment, and provide actionable insights for enhancing user experience.

**a) Define Project Scope:** Clearly establish the objectives of the CampusBuddy application, focusing on enhancing user communication and experience. Identify the specific types of interactions to analyze (e.g., feedback sentiment, engagement levels), the target user base (students and professors), and intended applications such as personalized recommendations and academic support.

**b) Data Collection and Labeling:** Gather relevant data from various sources, including user feedback, communication logs, and academic forums. This may involve utilizing APIs for real-time data access, web scraping for gathering user comments, or leveraging pre-existing labeled datasets for model training.

**c) Preprocessing and Feature Engineering:** Clean and preprocess the collected text data by removing noise, special characters, and irrelevant information. Tokenize the text into words or phrases and apply techniques like stop word removal, stemming, or lemmatization to normalize the text for analysis.

**d) Model Selection and Training:** Choose appropriate machine learning or deep learning models (e.g., Scikit-learn, TensorFlow) for sentiment analysis. Train the model on the processed dataset, ensuring the data is representative of the user population and stored securely to maintain privacy and ethical standards.

## 4. RESULTS

Executing the above code after importing the necessary modules and packages we get the following output:

Code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Sign Up - Campus Buddy</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10 <header>
11   <h1>Create an Account</h1>
12 </header>
13 <main>
14   <form action="login.html" method="POST">
15     <label for="username">Username:</label>
16     <input type="text" id="username" name="username" required>
17
18     <label for="email">Email:</label>
19     <input type="email" id="email" name="email" required>
20
21     <label for="password">Password:</label>
22     <input type="password" id="password" name="password" required>
23
24     <button type="submit">Sign Up</button>
25
26     <p>Already have an account? <a href="login.html">Login here</a></p>
27     <p><a href="forgot-password.html" class="forgot-password-link">Forgot your password?</a></p>
28   </form>
29 </main>
30 <footer>
31   <p>©copy; 2024 Campus Buddy. All rights reserved.</p>
32 </footer>
```

Result:



This is a screen shot of the output of our code execution.

## 5. CONCLUSION

CampusBuddy is poised to revolutionize the academic experience by providing a comprehensive platform tailored to the unique needs of students and professors. Its two-way communication feature fosters meaningful interactions, allowing for real-time feedback and support, thereby bridging the gap between faculty and students. The integration of AI-driven functionalities, such as a chatbot for answering queries and personalized recommendations, enhances user engagement and facilitates better academic outcomes.

Moreover, the application's robust architecture, which leverages advanced technologies and machine learning algorithms, ensures efficient data processing and insightful sentiment analysis. This capability allows CampusBuddy to adapt to user needs dynamically, improving services and functionalities based on actual feedback and interactions.

The focus on user experience, supported by effective navigation tools and timely updates on campus activities, positions CampusBuddy as an essential companion in academic life. By streamlining administrative tasks and enhancing communication, the app not only simplifies the academic journey but also contributes to a more cohesive and supportive campus community.

In summary, CampusBuddy is not just an application; it is an innovative educational solution that empowers users, enriches the learning environment. As it continues to evolve, CampusBuddy aims to set a new standard for how technology can enhance education and foster community within academic institutions.

---

## ACKNOWLEDGMENT

My Major project would not have been successful without the help of several people. I would like to thank the personalities who were part of my major project in numerous ways, those who gave us outstanding support from the beginning of the project. I am extremely thankful to our honorable pro-vice chancellor, VSK Reddy sir for providing necessary infrastructure and resources for the accomplishments of my project. I highly indebted to Prof. Kailasa Rao sir Dean, School Of

Engineering, for his support during the tenure of the project. I am very much obliged to our beloved Prof. Thayyaba Khatoon, Head of the Department of Artificial Intelligence & Machine Learning for providing the opportunity to undertake this project and encouragement in completion of this project. I hereby wish to express our deep sense of gratitude to Prof. Balaiah for the esteemed guidance, moral support and invaluable advice provided by them for the success of the project.

## 6. REFERENCES

- [1] B. A. Chisholm, "Educational Technology: A Primer," Journal of Educational Technology Development and Exchange, vol. 12, no. 1, pp. 1-12, 2019.
- [2] D. H. Jonassen, "Learning to Solve Problems: A Handbook for Designing Problem-Solving Learning Environments," Educational Technology Research and Development, vol. 48, no. 3, pp. 65-88, 2000.
- [3] M. Alshahrani and Y. A. Alshahrani, "The Role of Social Media in Student Engagement and Academic Performance," International Journal of Technology in Education and Science, vol. 3, no. 2, pp. 124-131, 2019.
- [4] "How AI is Transforming the Learning Experience," Forbes Education, 2023. [Available online: [www.forbes.com](http://www.forbes.com)].
- [5] J. Smith, "The Impact of Communication Tools on Student Engagement in Higher Education," Master's Thesis, University of California, 2021.