**Overview of backend technology**

Backend technology refers to tools, frameworks and languages used to build and maintain a backend system of applications.

By choosing the right backend technology for our project is a crucial decision, this is because it can influence its performance, scalability, security and overall success of the system as a whole.

* Briefly introduce the importance of choosing the right backend technology.
* Emphasize the impact on system functionality, scalability, and data privacy.

**Node.js**

Node.js is an ideal backend technology, with it having scalable network applications.

* Pros of using Node.js:
  + Developers can use this application for both frontend and backend development.
  + This application, encourages sharing of code, reduces the need to switch to between software’s for backend and frontend.
  + It has a rich package ecosystem via (Node package manager), which gives developers a range of pre-built modules and packages to use.
* Cons of using Node.js:
  + Due to it being a single-threaded nature it may limit CPU performance to perform intensive tasks.
  + Compared to more mature technologies, Node.js and its ecosystem are relatively young. This can sometimes result in fewer well-established patterns and best practices, leading to a learning curve for developers.

**Flask**

Flask, a microframework for python, which takes a minimalist approach to backend development, allowing developers to craft applications with only the components they need.

* Pros of using Flask:
  + Flask is extensible, allowing developers to use any third-party libraries and tools.
  + Flask uses the Jinja2 templating engine, providing a powerful and efficient way to render dynamic content. The template inheritance feature simplifies code organization and maintenance.
* Cons of using Flask:
  + It has less built-in functionality compared to larger frameworks such as Django. Meaning that developers may need to rely on external libraries for certain features.
  + Since Flask is suitable for small to medium-size applications, it may face scalability challenges in larger, more complex projects due to its microframework nature.

**Prioritizing data privacy and security**

For data protection, this includes things such as end-to-end encryption, secure communication protocols. To build user trust and follow the data protection regulations that are put into place.

**Node.js**

Single-threaded – Since it is single threaded event loop model architecture, it can handle multiple client requests, reducing the risks of the LSBU app from crashing. Single thread essentially executes the main loop while the input-output in the background is done by different threads.

* By having a look at, frontend and backend frameworks and taking the pros and cons into account, we have decided to use Node.js. Since, node.js can run on different operating systems, which would provide flexibility during development.
* This would ensure that the feature on the MyLSBU app can be run on different operating systems as needed. For example, windows and MacOS.
* Node.js is compatible with the major web browsers, such as, Google, Microsoft Edge, Firefox, Apple Safari etc.

The website that has been used: <https://alphacode.dev/backend/top-backend-technologies-2024/>

<https://www.geeksforgeeks.org/how-to-choose-the-right-backend-technology/>