Backend-Framework

Candidates	Pros	Cons
Django	• High-level (fast progress) • Robust Security • Community	• Aimed at large and complex projects • Less Async
Bottle	• Lightweight • Customizable • Minimalistic (Customizable)	• Scalability • Community • Minimalistic
FastAPI	• Learning curve • Flexible	• Community • Use case focus
Flask	• Lightweight • Customizable	• Scalability • Minimalistic

For U-Plant, our team ended the initial technological discussion with an emphasis on the use of Python (see section: python), several team members placed an added emphasis on security and disclosure as well. In the end we decided that Django would be best suited to our needs.

Django is implemented in python, so it initially showed promise. Other python frameworks include Bottle, Fast API, and Flask. Much like the latter three, Django offers robust security potential, but Django has these built in while the others require their seamless integration. In addition to its robust security features, unique to Django is an ORM, which will grant us the interface necessary to handle plant and user data with stride and focus on our project goals more directly. And while Django may traditionally be aimed at vastly large and complex projects, due to its alignment with our knowledge and values and its broad reaching out of box features, we have decided to leverage it in our app.

Frontend-Framework

Candidates	Pros	Cons
Tailwind CSS	• Highly customizable • Utility-First = Small CSS files • Responsive	• Steep learning curve • Cluttered HTML b/c lots of class names

Bootstrap	• Many components • Responsive • Large community	• Heavy framework • Common styles (less unique UI)
Materialize CSS	• Quick prototyping • Modern UI	• Less customizable • Small community • Limited third-party resources
Bulma	• Lightweight • Modern design • Easy learning	• Small community • Fewer components
Jinja	• Similar syntax to Python • Compatible with Python frameworks • Async support	• Third-party packages for frameworks often not supported

We chose Bootstrap as the best frontend framework for our project. It comes with a lot of ready-to-use components like navigation bars, modals, and forms, which speed up development. Its responsive grid system makes sure the app works well on different devices and screen sizes. Plus, Bootstrap has great documentation and community support, making it easier to troubleshoot and customize—important since we have a tight schedule and limited resources.

While Bootstrap is heavier and can lead to more basic-looking designs compared to Tailwind CSS, the benefits, like its reliability and ease of use, make it worth it. It helps keep the UI consistent and scalable, making future updates easier. Also, since our team is already familiar with Bootstrap, it reduces the learning curve and boosts productivity, making it the right choice for U-Plant.

Version Control

GitHub:

GitHub is a web-based platform for version control and collaboration that allows developers to store, track, and manage their code. It also provides tools for continuous integration, making it popular for open-source and private development projects.

GitLab:

GitLab is a web-based DevOps platform that provides version control, Continuous Integration/Continuous Deployment, and project management tools, all integrated into one system.

Bitbucket:

Bitbucket is a Git-based platform for version control and collaboration, similar to GitHub. It offers both cloud-hosted and self-hosted options, making it useful for managing code, tracking issues, and automating workflows.

Candidates	**Pros**	**Cons**
GitHub	• Code review features • All members have experience • Vast community support and extensive documentation. • Simple workflow and collaboration for teams using Python/Django.	• High complexity and may be overwhelming for the purposes of this project
GitLab	• Code review features • Offers the ability to self-host, giving full control over repositories and project data	• Not all members have experience • Less community support and documentation • Requires server resources and maintenance, which might not be practical or necessary for our project
Bitbucket	• Code review features	• High complexity and may be overwhelming for the

	• Direct Jira integration could significantly aid project management	purposes of this project • Not all members have experience • Less community support and documentation

When selecting a version control software for managing our community garden web app, U-Plant, we considered three major platforms: GitHub, GitLab, and Bitbucket. All three options provide significant code review features such as pull request reviewers and merge checks, making them suitable for remote collaboration among multiple coders and maintaining code quality. In the assessment, however, familiarity played the largest role in our decision. Since every member of the team already has significant experience with GitHub, we decided that it would be the most suitable platform for our needs.

IDE

Candidates	**Pros**	**Cons**
VS Code	- Great UI - Flexible - Versatility - Multi-language support	- Requires more configuration - Resource intensive
Anaconda	- Easy for beginners - Easy package manipulation	- Large size - Nobody in group is familiar with this IDE
PyCharm	- Solely focused on Python - Intuitive and simple - Refactoring tools	- Large size - Complex configuration - Resource intensive

When choosing between IDE's, we decided on VS Code. VS Code is more flexible and has an easy and intuitive UI. In addition, all group members are familiar with this IDE. Nobody is familiar with Anaconda, so we decided to throw that out. We're all familiar with PyCharm but as a group we decided that VS code would be easier to work with for this project.

Works Consulted

https://www.monocubed.com/blog/django-alternatives/

https://www.djangoproject.com/

https://flask.palletsprojects.com/en/3.0.x/

https://github.com/pricing

https://about.gitlab.com/features/?stage=plan

https://www.atlassian.com/software/bitbucket/pricing

https://www.git-tower.com/blog/git-hosting-services-compared/

https://www.webforefront.com/django/usejinjatemplatesindjango.html

https://medium.com/@ssc.ahmed.926748/what-are-the-advantages-and-disadvantages-of-using-visual-studio-code-or-atom-d3132bf1af85