



**Software Engineering
Bootcamp**

Hyperiondev

Building Secure Full Stack Applications

Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- ❑ No question is daft or silly - **ask them!**
- ❑ There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- ❑ You can also submit questions here:
<http://hyperiondev.com/sbc4-se-questions>
- ❑ For all non-academic questions, please submit a query:
www.hyperiondev.com/support
- ❑ Report a safeguarding incident:
<http://hyperiondev.com/safeguardreporting>
- ❑ We would love your feedback on lectures:
<https://hyperiondev.wufoo.com/forms/zsgv4m40ui4i0g/>

Objectives

1. Secure Full Stack
 - a. API
 - b. Security
 - c. Backend
 - d. Frontend

Github Repository – Lecture Examples/Slides

https://github.com/HyperionDevBootcamps/C4_SE_lecture_examples

Documentation

pytest: <https://docs.pytest.org/en>

Flask: <https://flask.palletsprojects.com/en/2.3.x/>

Flask testing: <https://flask.palletsprojects.com/en/2.3.x/testing/>

Postman: <https://www.postman.com/>

HTML: <https://devdocs.io/html/>

What is an API(Application Programming Interface)?

- Everything is an API
- An API is a way for computer programs to communicate.
- It is a type of software interface, offering a service to other pieces of software.

APIs

- We usually want our APIs to conform to some sort of architecture.
- We had a look at the rest architecture(Representational State Transfer)
 - Stateless
 - Uniform interface
 - Layered System
 - Cacheable

User Data

- When we allow the public to use our webapps we will probably require some information from them to provide a better experience.
- When we require this information there is a very high chance that we will get sensitive information from the user or just basic information that we would not want in the hand of malicious actors.
- We have to be careful about how we add data to our database and should also restrict who has access to this data.

Encryption and Hashing

- When we store user information we will always want to encrypt or hash the data.
- We would preferable hash passwords as it is seen as a one way street and is extremely difficult to reverse.
- It is important to encode all user data as a compromise on your system might compromise the user on another system as well.

Authentication

- To prevent users from accessing data that they should not, we want to have some sort of authentication.
- This will allow us to know who the user requesting the data is and if they have permission access to the data or not.
- By adding authentication we have an extra layer of protection to our data as the data is not only encrypted but only certain people have access to the data.

Backend

- We can accomplish our authentication with our back end.
- With our backend code we can retrieve the data from the correct destinations and do the necessary checks for authentication.
- Other than authentication the backend also allows us to add and store data to a database, add logic, build APIs and manage our servers.

Frontend

- The frontend is responsible for the user interface and interaction.
- Frontend developers use languages such as HTML, CSS and JavaScript
- They use HTML and CSS to provide style and a layout for the pages, then use JavaScript to add interaction to the elements.

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any



Hyperiondev

Thank you for joining us

Stay hydrated
Avoid prolonged screen time
Take regular breaks
Have fun :)