**Thema**

*Die Themenformulierung darf nicht aus nur einem Wort bestehen und sollte 100 Zeichen nicht überschreiten.*

*Nach der Genehmigung des Themas durch die Schulbehörde ist dieses nicht mehr veränderbar. Die genehmigte Formulierung wird ins Reifeprüfungszeugnis übernommen.*

*Die fertiggestellte Arbeit kann einen Untertitel enthalten.*

Monolithisch oder Microservices? Ein Vergleich anhand der Frameworks Flask und Django

Monolithic vs. Microservices: a comparison based on the frameworks Flask and Django

**Inhaltliche Zuordnung**

Naturwissenschaftlicher Bereich, Mathematik, Informatik

**Sprache der Arbeit**

*Wenn die Arbeit in einer Fremdsprache verfasst wird, ist das Thema auf Deutsch und in der Sprache der Arbeit anzugeben.*

*Die Felder des Erwartungshorizonts sind in jedem Fall auf Deutsch auszufüllen.*

English

**Erwartungshorizont**

**Persönlicher Impuls und erste Basisliteratur**

*Führen Sie in zwei bis drei Sätzen Ihre Gründe für die Wahl des Themas an! Nennen Sie einige Bücher, Internetseiten, Filme oder andere Medien, die Sie bei der ersten Einarbeitung in Ihr Thema benutzt haben. Anzugeben sind jeweils Autor, Titel, Erscheinungsjahr und bei Online-Ressourcen zusätzlich die Internetadresse (mit Datum des letzten Zugriffs)!*

*Maximal 1500 Zeichen*

Ich habe dieses Thema gewählt, da ich mich selber ein bisschen mit Webdevelopment auseinandergesetzt habe und aus eigener Erfahrung die Vor- und Nachteile der beiden Arten kennengelernt habe.

Literatur:

Forcier J., Bissex P., Chun W. (2008) Python Web Development with Django, Addison-Wesley Professional

Bibeault B., Katz Y., De Rosa A. (2015) jQuery in Action, 3rd edition, Manning

Newman S. (2015) Building Microservices: Designing Fine-Grained Systems, O’Reilly Media Inc.

Webber J., Parastatidis S., Robinson I. (2010) REST in Practice, O'Reilly

Internet Seiten:

Abrufdatum (23.02.2017 15:08) gilt für alle folgenden Links

Brail G., Building APIs with Swagger <<http://radar.oreilly.com/2015/09/building-apis-with-swagger.html>>

Zalando (2015) Swagger/OpenAPI First framework for Python on top of Flask <<https://github.com/zalando/connexion>>

Django Software Foundation, Django <<https://www.djangoproject.com>>

Flask <<http://flask.pocoo.org>>

Python <<https://www.python.org>>

Python full stack <<http://www.fullstackpython.com>>

**Geeignete Leitfragen**

*Was möchten Sie herausfinden? Was interessiert Sie am gewählten Thema besonders? (Eine Konkretisierung bzw. Adaptierung der Leitfrage/n bzw. Fragestellung/en ist im Verlauf der weiteren Auseinandersetzung mit dem Thema möglich.)*

*Maximal 500 Zeichen*

Wie sind die beiden Systeme aufgebaut?

Was sind die wesentlichen Unterschiede?

Wie wirkt sich die jeweilige Methode auf den Server, wie auf den Client aus? (Serverlast, Clientlast)

Aus AnwenderInnensicht: Wie wirkt sich die gewählte Methode auf User Experience und Usability aus?

Aus EntwicklerInnensicht: Wie ist Aufwand, Wartbarkeit und Skalierbarkeit der jeweiligen Methode?

Was sind die Vor- und Nachteile der beiden Frameworks? Was sind die wesentlichen Unterschiede?

**Angestrebte Methode/n**

*Soll eine reine Literaturarbeit verfasst werden oder soll die Arbeit auch empirische Elemente (naturwissenschaftliche Versuchsanordnungen, Fragebogenerhebungen, Programmiertätigkeit etc.) enthalten?*

*Maximal 500 Zeichen*

Literaturarbeit mit Programmiertätigkeit

**Ungefähre Gliederung der Arbeit**

*Listen Sie die inhaltlichen Schwerpunkte Ihrer Arbeit stichwortartig in der voraussichtlichen Reihenfolge auf!*

*Maximal 500 Zeichen*

- Kurze Einführung in die Thematik  
- Programmieren einer möglichst simplen Applikation mithilfe des Python-Frameworks Flask  
- Analyse des geschriebenen Programms  
- Programmieren derselben Applikation mit dem Python-Framework Django  
- Analyse des geschriebenen Programms  
- Vergleich der Vor- und Nachteile und Conclusio