**Aufgabe: Kontaktformular mit HTTP Post und Templating**

**Anforderungen:**

1 Start-Seite mit einem beliebigen Seiten-Titel

* Besitzt 2 Buttons, die auf andere Seiten verweisen (weiter unten beschrieben):
  + "Anfrage"
  + "Kontaktanfragen"

1 User-Seite mit einem Kontaktformular mit dem Seiten-Titel "Anfrage":

* Eingabe von Betreff (Einzeiliger Freitext, verpflichtend)
* Eingabe von Name (optional)
* Eingabe von einer Antwort-Email-Adresse (verpflichtend, Email-Format soll so weit möglich überprüft werden - serverseitige Validierung ist optional)
* Eingabe einer Nachricht (Mehrzeilger Freitext, verpflichtend)
* Absenden-Button
* Beim Absenden ist auch Datum und Uhrzeit der Anfrage abzuspeichern
* Nach dem Absenden soll der User weitergeleitet werden auf die nächste Seite "Bestätigung"
* Ein Button, der zur Start-Seite zurückführt

1 User-Seite als Bestätigung mit dem Seiten-Titel "Bestätigung":

* Anzeige eines statischen Texts "Anfrage abgeschickt"
* Ein Button, der zur Start-Seite zurückführt
* F5/Ctrl+R soll hier KEINE Browsermeldungen verursachen

1 Admin-Seite mit einer Übersicht über eingegangene Kontaktanfragen mit dem Seiten-Titel "Kontaktanfragen":

* Listenansicht der Anfragen, sortiert nach Datum+Uhrzeit, mit den neuesten Anfragen zuerst
* Jeder Listeneintrag soll die folgenden Informationen tragen
  + Datum und Uhrzeit der Anfrage
  + Name (sofern vorhanden)
  + Antwort-Email-Adresse
  + Betreff
* Einzelne Anfragen können in einer weiteren Seite geöffnet werden
* Ein Button, der zur Start-Seite zurückführt

1 Admin-Seite zur Anzeige einzelner Anfragen:

* Anzeige des Nachrichteninhalts
* Besitzt 2 Buttons, die auf eine andere Seite führen:
  + zurück zur Start-Seite
  + zurück zu "Kontaktanfragen"
* HTML escaping beachten!

**Implementierungshinweise**

* Die 5 Seiten sollen jeweils mit einem jinja-Template implementiert sein
* Anfragen von Usern sollen per HTTP POST geschickt werden
* Laden, speichern und auflisten der Anfragen soll in einer storage-klasse implementiert sein
* Die 5 Endpunkte sollen mit unit tests versehen werden
* Adressierung der einzelnen Anfragen zur Anzeige kann über Pfad-Variablen in FastAPI gelöst werden: <https://fastapi.tiangolo.com/#example>

@app.get("/items/{item\_id}")

def read\_item(item\_id: int):

*# Ein HTTP GET Call zu /items/8 ruft diese funktion mit item\_id = 8 auf*

pass