**QuizNinja API – CS310 Backend Demo Report**

**Ahmet Alperen Güngör – Anıl Arslan**

Quiz Ninja API is a simple API to display questions added by admins and students, allowing students to solve questions & see their solutions, while commenting and discussing the question with the other users.

**Endpoints:**

* User Registration:
  + Description: Register a user to the database with given username and password
  + Address: <http://localhost:8080/api/register>
  + Method: POST
  + Body: JSON file consisting username and password

{

"username" : "",

"password" : ""

}

* + Returns:
    - If username already exists, return “Username already exists” exception
    - If not, JSON file consisting user information

{

"id": "6574fd4646421d715f80e863",

"username": "reportUser",

"password": "pass123"

}

* User Login:
  + Description: Login a user to the database with given username and password
  + Address: http://localhost:8080/api/login
  + Method: POST
  + Body: JSON file consisting username and password

{

"username" : "",

"password" : ""

}

* + Returns:
    - If username does not exist in database, return “Username is wrong”
    - If username exist but password does not match, return “Password is wrong”
    - If credentials correct, JSON file consisting user information

{

"id": "6574f9d586d4a472c4e8ec3c",

"username": "myUser",

"password": "pass123"

}

* Posting questions:
  + Description: In addition to solving questions, students can add questions with solutions as well
  + Address: http://localhost:8080/api/postQuestion
  + Method: POST
  + Body: JSON file with question information (Source is set to student if added by student)

{

"question\_url": "",

"solution\_url": "",

"course": "",

"source": "",

"comments" : []

}

* + Returns: Same information if correctly added
* Getting all questions:
  + Description: Return all questions in database to display (Frontend will parse course type)
  + Address: http://localhost:8080/api/questions
  + Method: GET
  + Returns: JSON file consisting of all question objects

[

{

"id": "6574f6b51694a0256d178d05",

"question\_url": "myurl",

"solution\_url": "mysolutionurl",

"course": "Matematik",

"source": "Database",

"comments": []

},

{

"id": "6574fa0886d4a472c4e8ec3d",

"question\_url": "myquestionimage",

"solution\_url": "mysolutionimage",

"course": "Fen",

"source": "Database",

"comments": []

}

.

.

.

]

* Writing a comment for a question:
  + Description: Users can comment & discuss on questions
  + Address: http://localhost:8080/api/comment/{question-id}/{user-name}
  + Method: POST
  + Body: Comment text in JSON file

{

"text" : ""

}

* + Returns: JSON file consisting of question object that was commented on with comment added

{

"id": "6574e87fc4d1d169139653a8",

"question\_url": "myurl”,

"solution\_url": "myurl2",

"course": "Matematik",

"source": "Database",

"comments": [

{

"username": "myUser",

"text": "{\r\n \"text\" : \" hi demo 2\"\r\n}"

}

]

}