A.

URLS –

a. /quizzes – To GET all quizzes stored in the database
b. /quizzes/past – To GET all the past quizzes
c. /quizzes/live – To GET all the live quizzes
d. /quizzes/upcoming – To GET all the upcoming quizzes
e. /quizzes/create – Admin can create a quiz by providing title, start date-time and end-date time. Below is an example of JSON object passed in a POST request –

```
{
    "title": "Quiz Title",
    "start_time": "2021-06-24T20:00:00+05:30",
    "end_time": "2021-06-24T23:00:00+05:30"
}
```

f. /quizzes/<quiz_id>/addMCQ – To add MCQ question to the database, belonging to quiz with <quiz_id>
g. /quizzes/<quiz_id>/addTextual – To add Textual question to the database, belonging to quiz with <quiz_id>
h. /quizzes/<quiz_id>/questions – To GET all the questions of a particular quiz with <quiz_id>
i. /quizzes/<quiz_id>attempt –
    a. GET request
        i. Returns Response of all questions of quiz with <quiz_id>.
        ii. Stores a record in the Score table with the request.user.username and <quiz_id>. Score is set to by default 0. Submitted is set to false, until a POST request with answers is provided by the same user.
    b. POST request -
    Below is an example of how answers to the quizzes should be posted on the API server.

```
{
    "MCQ":[
      {
        "id": 2,
        "user_answer": 4
      },
      {
        "id": 4,
        "user_answer": 2
      }
    ],
    "Textual":[
      {
        "id":1,
        "user_answer":"No Answer"
      },
      {
        "id":2,
        "user_answer":"This is the answer"
      }
    ]
}
```

Calculates the score of the user based upon the answer provided, and set submitted field in Score table to True.

B.

We need to consider the following points while designing the API server to handle 1 million users accessing the live quizzes at the same time –

1. OS of the server system – as in Linux or Windows. Linux servers are usually faster in handling requests than Windows server.
2. Providing CDNs and a greater number of servers can improve speed of handling requests.
3. Good caching systems like Nginx for handling requests from cache can help reduce burden on web servers and improve the speed at which API servers will handle requests.