Answers

1. Let’s assume the server is multithreaded and contains a database with the variable column “visit count” which sums the number of times that each redirection was used. For the server to send frequently accessed faster we need to make a special thread that creates a new small database in conjunction with the main database. This thread will monitor the main database for the most used URLs (let’s say top 10 URLs) using an ORDER BY command and then Insert them in the smaller database for faster access. So that when a redirection occurs the server first checks if the database which contains the most used URLs and then the rest.

2. To implement a “one short UR few long URLs” system we will need to make integer array where each cell represents the amount of redirection for each URL based on the percentage that user set for each URL, then we check if the percentage of redirects to a certain URL is less than the needed if its less we redirect to the URL and add 1 to the amount of redirects that it has if not we move to

another URL. see the example code below for a better understanding of the concept

```
Counts = [0, 0]
Percentages = [80, 20]
URLs = ["asd", "abc"]
i = 0
if i >= len(URLs):
     i = 0
if Percentages[i] >= ((Counts[i]/sum(Counts) if sum(Counts) else 0) * 100):
     redirect(URLs[i])
     Counts[i] += 1
else:
     i += 1
```

3. Much like the answer to question 1 a special thread is made with a special database that this time the amount and the time are a factor to check, in the project there’s a bonus database that logs the time and the url of each redirection. So to find the top 5 most used URLs in the last Hour we’ll have to select all of the redirection that were made in the last hour and and then we use COUNT and GROUP BY to count the number of redirects of each URL. Then we select the first 5 and save them in the database. We update each time there a redirection.

4. The described attack can be classified as a DOS attack, and for such an attack to be prevented there’s a need for a packet scanner i.e a function that checks the amount of packets that come from the same IP in second, if the amount is more than lets say 5 then it’s an attack and the user will be black listed
   In reality thought by the time the packet scanner receives the data it’s too late as the server will be exhausted from resources, a safer way to defend from these attacks is through hardware

5. When the user presses the Create button a function is being triggered which checks if the input is empty or not, if empty an error will display and if not a fetch request will be made with the data being the input in JSON format and sent this request to the server, the server will later receive the data and convert it from JSON to a basic string, then the server will check if the URL is already in the database and if not it will call a function that will make a short URL for the redirection, afterwards the server will add a new line to the server with the original URL and short one and then will send the short URL back to the user, back at the front-end the fetch request also

receives data from the server, but this time in plain text because of the simplicity of the short url (for not having complex characters) and then it outputs the string to the webpage