[Flask]

General Information & Licensing

Code Repository	https://github.com/pallets/flask
License Type	BSD-3-Clause license
License Description	• Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. • Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. • Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
License Restrictions	• It prohibits the names of the authors from being used to endorse or promote products relating to the software. • It prohibits others from using the name of the copyright holder or its contributors to promote derived products without written consent. • This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. in no event shall the copyright holder or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage. (liability and warranty)

Magic

In a flask frame, basically it uses a port to establish TCP connection. It means that it will listen on a port on a TCP socket, after it receives an HTTP request via that port, then it will parse the data according to protocol format. Here is the screenshot of the code that used in our code (line 131-133) :

app.py

```
109         users_test_account.insert_one({"username":username, "password":password_se, "salt":salt, "won":"0", "games":"0"})
110         feedback = {"status": "True", "username": username}
111         emit('signup',feedback)
112
113 @socketio.on("create", namespace="/")
114 def create(message):
115     print(message)
116     message["room_num"] = str(random.randint(0, 100))
117     emit('create', json.dumps(message))
118
119
120
121
122 # def password_confirm(username, input_password):
123 #     for user in game_engine.users_info:
124 #         if user["username"] == username:
125 #             return user["password"] == input_password
126 #     return False
127
128
129
130
131 if __name__ == '__main__':
132     # from waitress import serve
133     app.run(host="0.0.0.0", port=5000)
134     # socketio.run(app)
135
136
137
138
139
```

It shows that the port 5000 was used to listen for a TCP socket, after the TCP is established via this port, then the HTTP request can be sent from client to server and response can be sent from server to client.

In this function, it consists of two methods to establish TCP connection. The first one is app.run(), which is between line 1067 and 1074 in src/flask/app.py (https://github.com/pallets/flask/blob/main/src/flask/app.py)

```
1067     def run(
1068         self,
1069         host: t.Optional[str] = None,
1070         port: t.Optional[int] = None,
1071         debug: t.Optional[bool] = None,
1072         load_dotenv: bool = True,
1073         **options: t.Any,
1074     ) -> None:
```

In the run function, it is used to run the application in the local development. It means that when we specific every argument that we need in our project, it can run the application according to the specific argument.

The port is used to specify the port of the server to listen to the HTTP request. In our code, it

set to 5000, then after run function is executed, the application should be established in the local development, and then the port 5000 is used to establish TCP connection.