We have solved the miniban project in the following way: our files are miniban.sh, ban.sh, unban.sh, failIP.txt, miniban.db and whitelist.db. Each script has comments explaining the important parts of the code; however, we will still explain some of it here.

Miniban.sh runs and starts by checking whether the user is root. If the user is root, the script will continue, but if the user is not root, the script will end. We start by checking this because some of the commands require root privileges, and there is no point in running the script without them. We use a while-loop to constantly watch for failed and successful ssh login attempts. If a new failed login is detected, the IP address will be placed in a temporary text file. The next line in our script uses grep to find the amount of failed logins from this IP address. If the same IP address has failed 3 times the ban.sh script runs. After we have watched for failed logins, we watch for new successful logins. If a new successful login is detected, we retrieve that IP and remove matching IP addresses from the failIP.txt document. The purpose is to allow a person to log in on their second or third attempt, and still have three chances on their next login (the same as a person who is banned and then unbanned).

Ban.sh starts by setting the variable IP to what it received from miniban.sh. This IP is then compared against the whitelist database. If there is a match, the script will stop running, however, if there is no match in the whitelist database, iptables is used to ban the IP from trying to log in again. The ban is logged in miniban.db with the format IP, TIMESTAMP, so that unban.sh can read when the IP was banned and unban after ten minutes.

Unban.sh runs every loop in the while-loop in miniban. In unban.sh we check whether there are any banned IP addresses in miniban.db. If we find an IP there, unban.sh will continuously check if more than 10 minutes have passed since the IP was banned. When more than 10 minutes have passed, the IP address will be unbanned and removed from the database. This process is done for every IP address in miniban.db. The way our program is written we need an else-sentence for if [$MIN -ge 10]. We need this because otherwise, the program will get stuck unbanning one IP and not detect other IP addresses that try to log in between banning the first IP and the unbanning of the first IP.

The remaining files are for storage. FailIP.txt is temporary for current failed login attempts. Miniban.db stores banned IP addresses and whitelist.db stores whitelisted IP addresses.