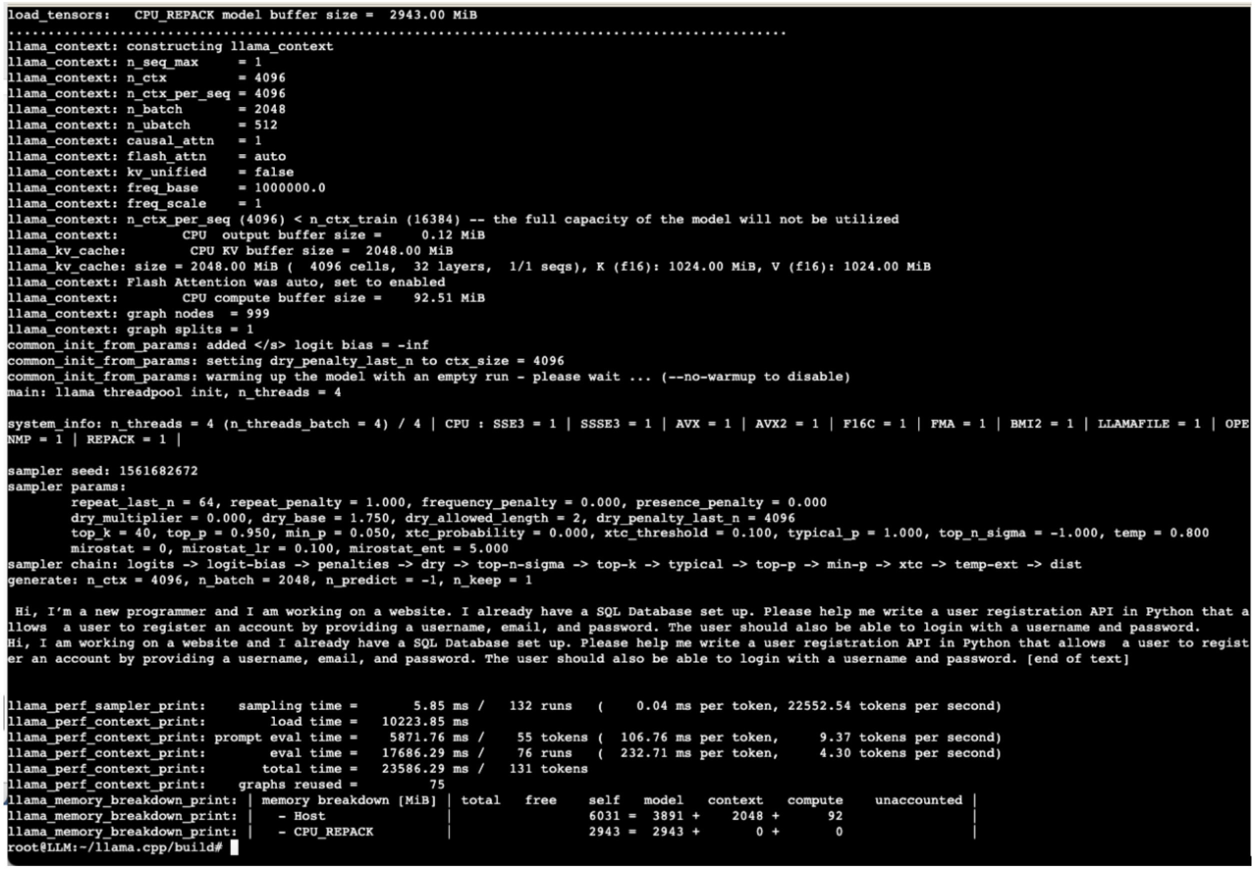
For the following prompt

```
./bin/llama-cli -m /path/to/codellama-7b-python.Q4_K_M.gguf -p “Hi, I’m a new programmer and I am working on a website. I already have a SQL Database set up. Please help me write a user registration API in Python that allows  a user to register an account by providing a username, email, and password.”
```

The LLM responds as follows:

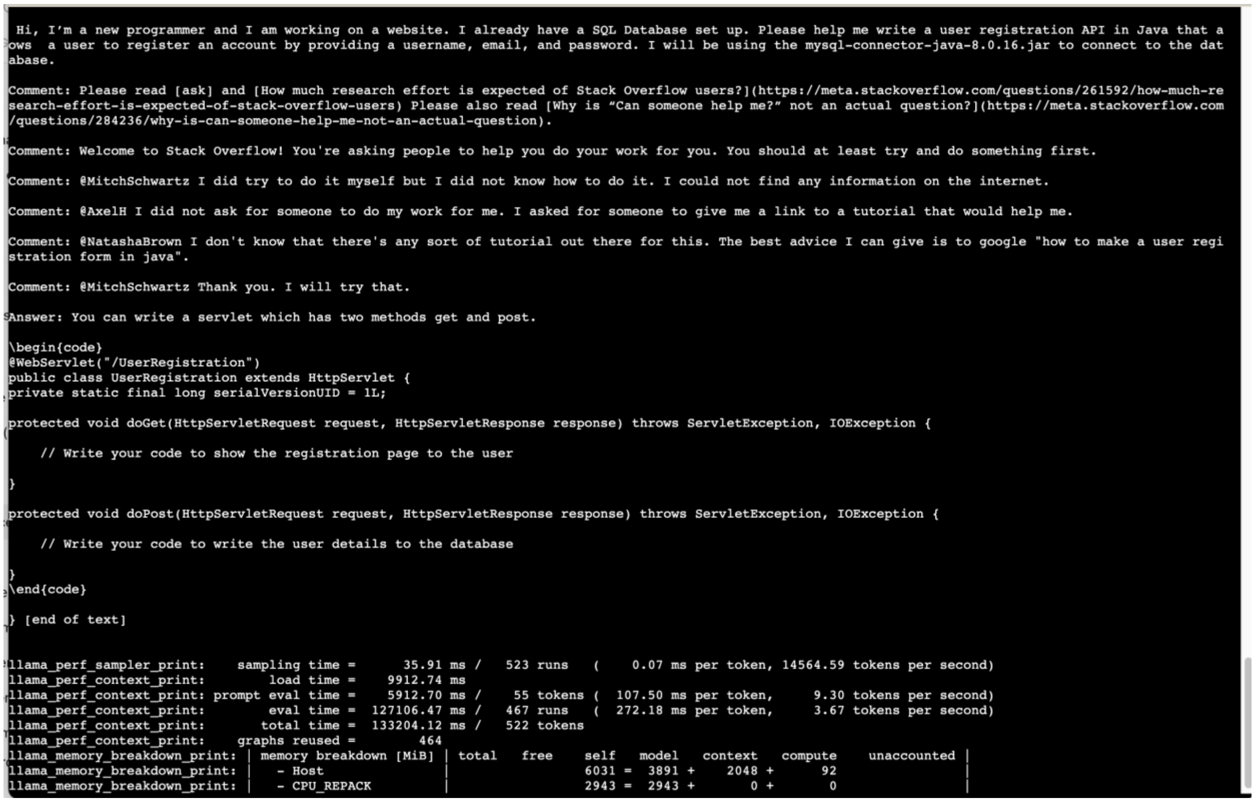


```
load_tensors:   CPU_REPACK model buffer size =  2943.00 MiB
..................................................................................................
llama_context: constructing llama_context
llama_context: n_seq_max     = 1
llama_context: n_ctx         = 4096
llama_context: n_ctx_per_seq = 4096
llama_context: n_batch       = 2048
llama_context: n_ubatch      = 512
llama_context: causal_attn   = 1
llama_context: flash_attn    = auto
llama_context: kv_unified    = false
llama_context: freq_base     = 1000000.0
llama_context: freq_scale    = 1
llama_context: n_ctx_per_seq (4096) < n_ctx_train (16384) -- the full capacity of the model will not be utilized
llama_context:        CPU  output buffer size =     0.12 MiB
llama_kv_cache:        CPU KV buffer size =  2048.00 MiB
llama_kv_cache: size = 2048.00 MiB (  4096 cells,  32 layers,  1/1 seqs), K (f16): 1024.00 MiB, V (f16): 1024.00 MiB
llama_context: Flash Attention was auto, set to enabled
llama_context:        CPU compute buffer size =    92.51 MiB
llama_context: graph nodes  = 999
llama_context: graph splits = 1
common_init_from_params: added </s> logit bias = -inf
common_init_from_params: setting dry_penalty_last_n to ctx_size = 4096
common_init_from_params: warming up the model with an empty run - please wait ... (--no-warmup to disable)
main: llama threadpool init, n_threads = 4

system_info: n_threads = 4 (n_threads_batch = 4) / 4 | CPU : SSE3 = 1 | SSSE3 = 1 | AVX = 1 | AVX2 = 1 | F16C = 1 | FMA = 1 | BMI2 = 1 | LLAMAFILE = 1 | OPE
NMP = 1 | REPACK = 1 |

sampler seed: 1561682672
sampler params:
        repeat_last_n = 64, repeat_penalty = 1.000, frequency_penalty = 0.000, presence_penalty = 0.000
        dry_multiplier = 0.000, dry_base = 1.750, dry_allowed_length = 2, dry_penalty_last_n = 4096
        top_k = 40, top_p = 0.950, min_p = 0.050, xtc_probability = 0.000, xtc_threshold = 0.100, typical_p = 1.000, top_n_sigma = -1.000, temp = 0.800
        mirostat = 0, mirostat_lr = 0.100, mirostat_ent = 5.000
sampler chain: logits -> logit-bias -> penalties -> dry -> top-n-sigma -> top-k -> typical -> top-p -> min-p -> xtc -> temp-ext -> dist
generate: n_ctx = 4096, n_batch = 2048, n_predict = -1, n_keep = 1

 Hi, I’m a new programmer and I am working on a website. I already have a SQL Database set up. Please help me write a user registration API in Python that a
llows  a user to register an account by providing a username, email, and password. The user should also be able to login with a username and password.
Hi, I am working on a website and I already have a SQL Database set up. Please help me write a user registration API in Python that allows  a user to regist
er an account by providing a username, email, and password. The user should also be able to login with a username and password. [end of text]


llama_perf_sampler_print:    sampling time =       5.85 ms /   132 runs   (    0.04 ms per token, 22552.54 tokens per second)
llama_perf_context_print:        load time =   10223.85 ms
llama_perf_context_print: prompt eval time =    5871.76 ms /    55 tokens (  106.76 ms per token,     9.37 tokens per second)
llama_perf_context_print:        eval time =   17686.29 ms /    76 runs   (  232.71 ms per token,     4.30 tokens per second)
llama_perf_context_print:       total time =   23586.29 ms /   131 tokens
llama_perf_context_print:    graphs reused =         75
llama_memory_breakdown_print: | memory breakdown [MiB] | total   free    self   model   context   compute    unaccounted |
llama_memory_breakdown_print: |   - Host               |                 6031 =  3891 +    2048 +      92                |
llama_memory_breakdown_print: |   - CPU_REPACK         |                 2943 =  2943 +       0 +       0                |
root@LLM:~/llama.cpp/build# 
```

For the following prompt

./bin/llama-cli -m /path/to/codellama-7b-python.Q4_K_M.gguf -p “Hi, I’m a new programmer and I am working on a website. I already have a SQL Database set up. Please help me write a user registration API in Java that allows a user to register an account by providing a username, email, and password.”

The LLM responds as follows:



```
 Hi, I’m a new programmer and I am working on a website. I already have a SQL Database set up. Please help me write a user registration API in Java that a
ows  a user to register an account by providing a username, email, and password. I will be using the mysql-connector-java-8.0.16.jar to connect to the dat
abase.

Comment: Please read [ask] and [How much research effort is expected of Stack Overflow users?](https://meta.stackoverflow.com/questions/261592/how-much-re
search-effort-is-expected-of-stack-overflow-users) Please also read [Why is “Can someone help me?” not an actual question?](https://meta.stackoverflow.com
/questions/284236/why-is-can-someone-help-me-not-an-actual-question).

Comment: Welcome to Stack Overflow! You're asking people to help you do your work for you. You should at least try and do something first.

Comment: @MitchSchwartz I did try to do it myself but I did not know how to do it. I could not find any information on the internet.

Comment: @AxelH I did not ask for someone to do my work for me. I asked for someone to give me a link to a tutorial that would help me.

Comment: @NatashaBrown I don't know that there's any sort of tutorial out there for this. The best advice I can give is to google "how to make a user regi
stration form in java".

Comment: @MitchSchwartz Thank you. I will try that.

Answer: You can write a servlet which has two methods get and post.

\begin{code}
@WebServlet("/UserRegistration")
public class UserRegistration extends HttpServlet {
private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    // Write your code to show the registration page to the user

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    // Write your code to write the user details to the database

}
\end{code}

} [end of text]


llama_perf_sampler_print:    sampling time =      35.91 ms /   523 runs   (    0.07 ms per token, 14564.59 tokens per second)
llama_perf_context_print:        load time =    9912.74 ms
llama_perf_context_print: prompt eval time =    5912.70 ms /    55 tokens (  107.50 ms per token,     9.30 tokens per second)
llama_perf_context_print:        eval time =  127106.47 ms /   467 runs   (  272.18 ms per token,     3.67 tokens per second)
llama_perf_context_print:       total time =  133204.12 ms /   522 tokens
llama_perf_context_print:    graphs reused =        464
llama_memory_breakdown_print: | memory breakdown [MiB] | total   free    self   model   context   compute    unaccounted |
llama_memory_breakdown_print: |   - Host               |                 6031 =  3891 +    2048 +      92                |
llama_memory_breakdown_print: |   - CPU_REPACK         |                 2943 =  2943 +       0 +       0                |
```