МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени Н.Э.Баумана

ОТЧЕТ

о прохождении производственной практики

Место прохождения практики: «разработка Веб-сайта»

Студент Нгуен Тхань Дат

1 курс, учебная группа ИУ9-22Б

Руководитель практики:

Директор:

Москва 2019

Содержание

Введение, постановка задачи

Описание Веб-сайта

Описание работ, выполненных в период практики

1 этап. Front-end

2 этап. Back-end

Заключение

Введение, постановка задачи

Node или Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. (Wikipedia)

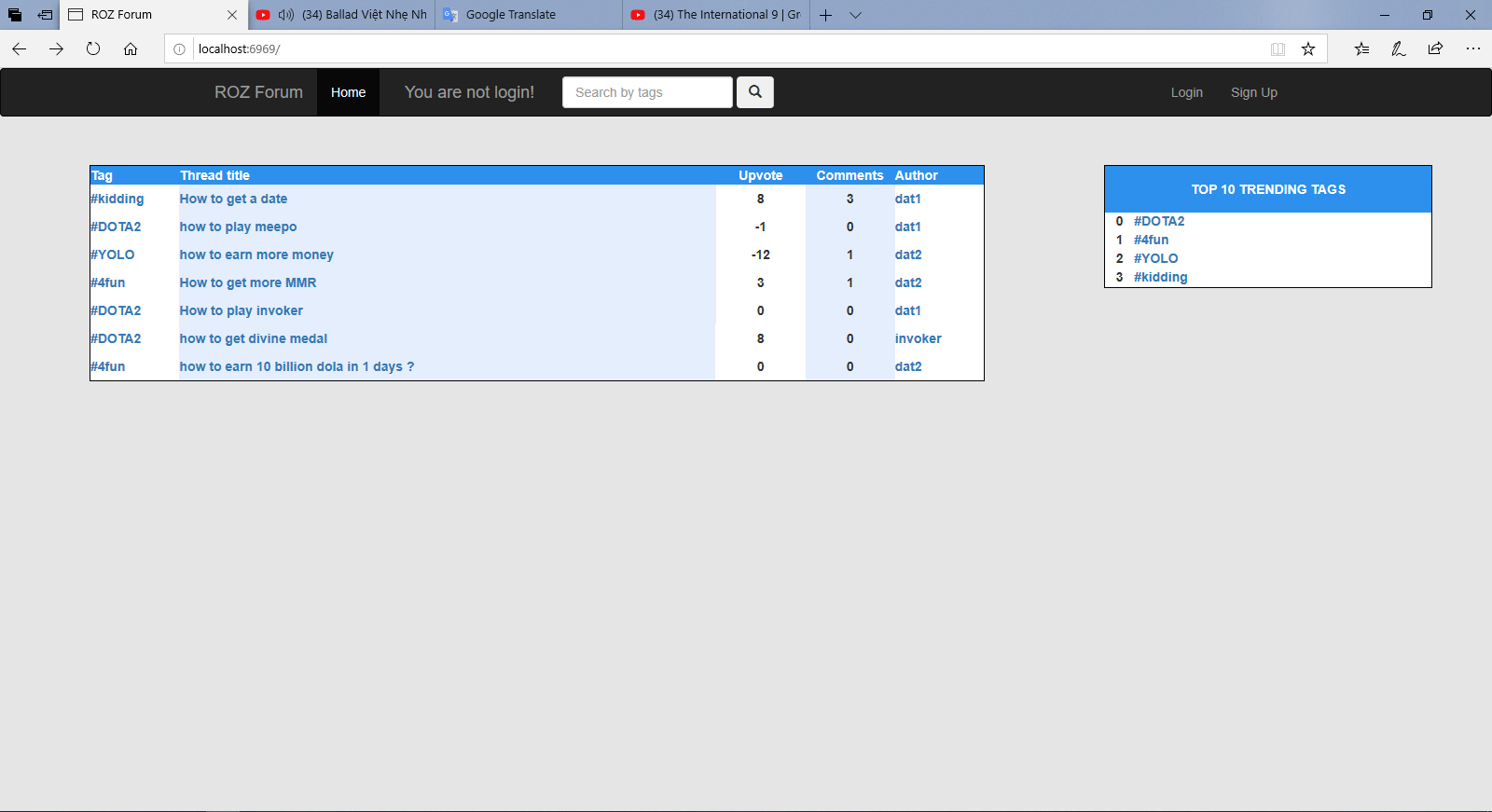
Описание проблемы: Создайте форум (веб-сайт) только с Javascript. Не используя Library или Framework. Форум (сайт) состоит из 2 частей. Первый - это Front-end, который написан на HTML, CSS и использует Javascript для доступа к HTML элементу - DOM. Второй - это Back-end, который использует NodeJS, написанный Javascript. Back-end создал две основные идеи: создание «HTTP Web Server» с помощью модуля “http” в NodeJS и использование файла JSON в качестве базы данных. Для удобства проверки я не использую никакой защиты для базы данных, аутентификации, Salting hasher method...

Описание работ, выполненных в период практики

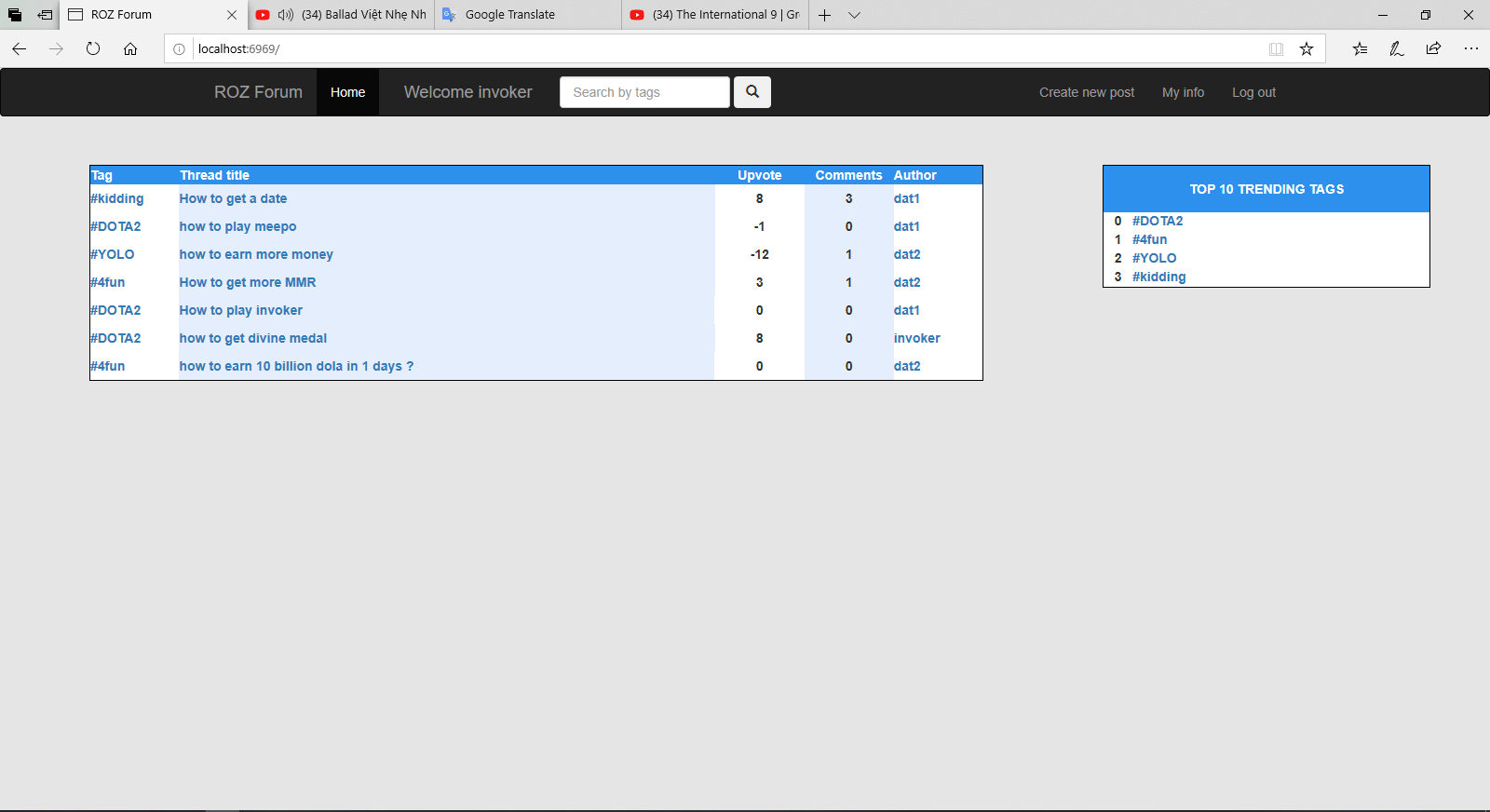
I: Front-end (UI / UX для веб-сайта)

1: Главная страница

до входа пользователя

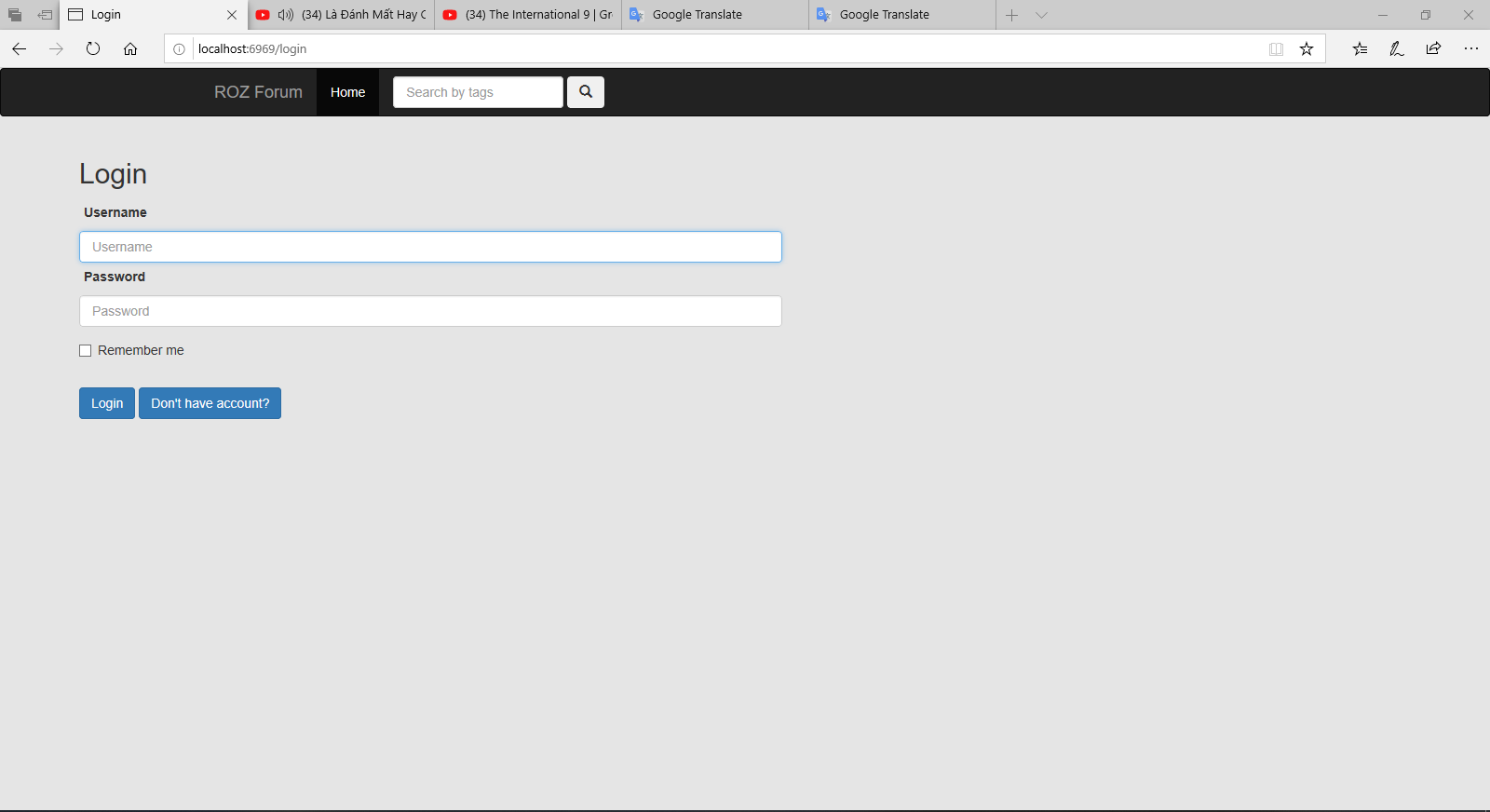


после входа пользователя



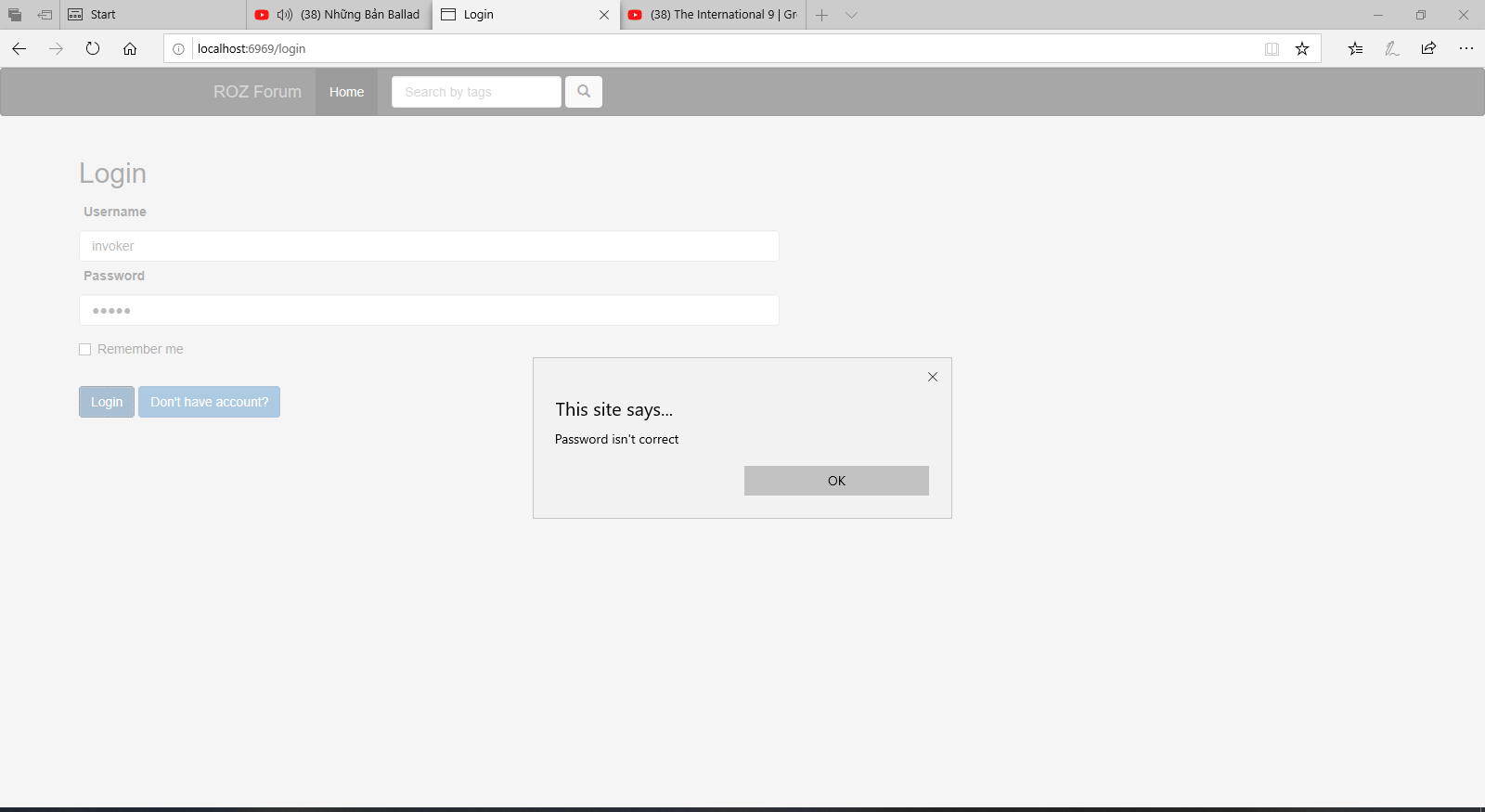
* В верхней части сайта есть кнопка перенаправления на главную страницу. Далее идет уведомление, оно будет меняться при входе пользователя в систему или нет. Далее это место для пользователя, ищущего тему по определенному тегу. Если пользователь не авторизуется, отобразятся 2 кнопки «Войти» и «Зарегистрироваться». При входе пользователя в систему будут отображаться кнопки «Создать новую тему», «Информация о пользователе», «Выйти из системы».
* На главной странице есть 2 таблицы
  + Первый показывает все темы, написанные пользователем. Каждый ряд - это каждый тем. Первый столбец является тегом тем. При нажатии будет перенаправлен на страницу, на которой показаны все темы с этим тегом. Второй столбец - это заголовок темы, щелкающий для отображения всей этой темы (содержимое, комментарии, ...). Следующий столбец - «upvote», если кому-либо нравится тем, «upvote» будет увеличиваться, в противном случае «upvote» будет уменьшаться. Следующий столбец - количество комментариев в этой теме. Последний столбец - автор, который пишет эту тему. Нажав, чтобы показать эту информацию пользователя.
  + Второй показывает топ 10 трендовых тегов. Нажав на каждый тег, чтобы показать все темы с этим тегом.
* Kак это устроенo
  + Используя HTML и CSS для создания и организации
  + Используя JavaScript для доступа к HTML элементу – DOM
  + Используя xmlHttpRequest для доступа к базе данных.

2: Страница для входа пользователя

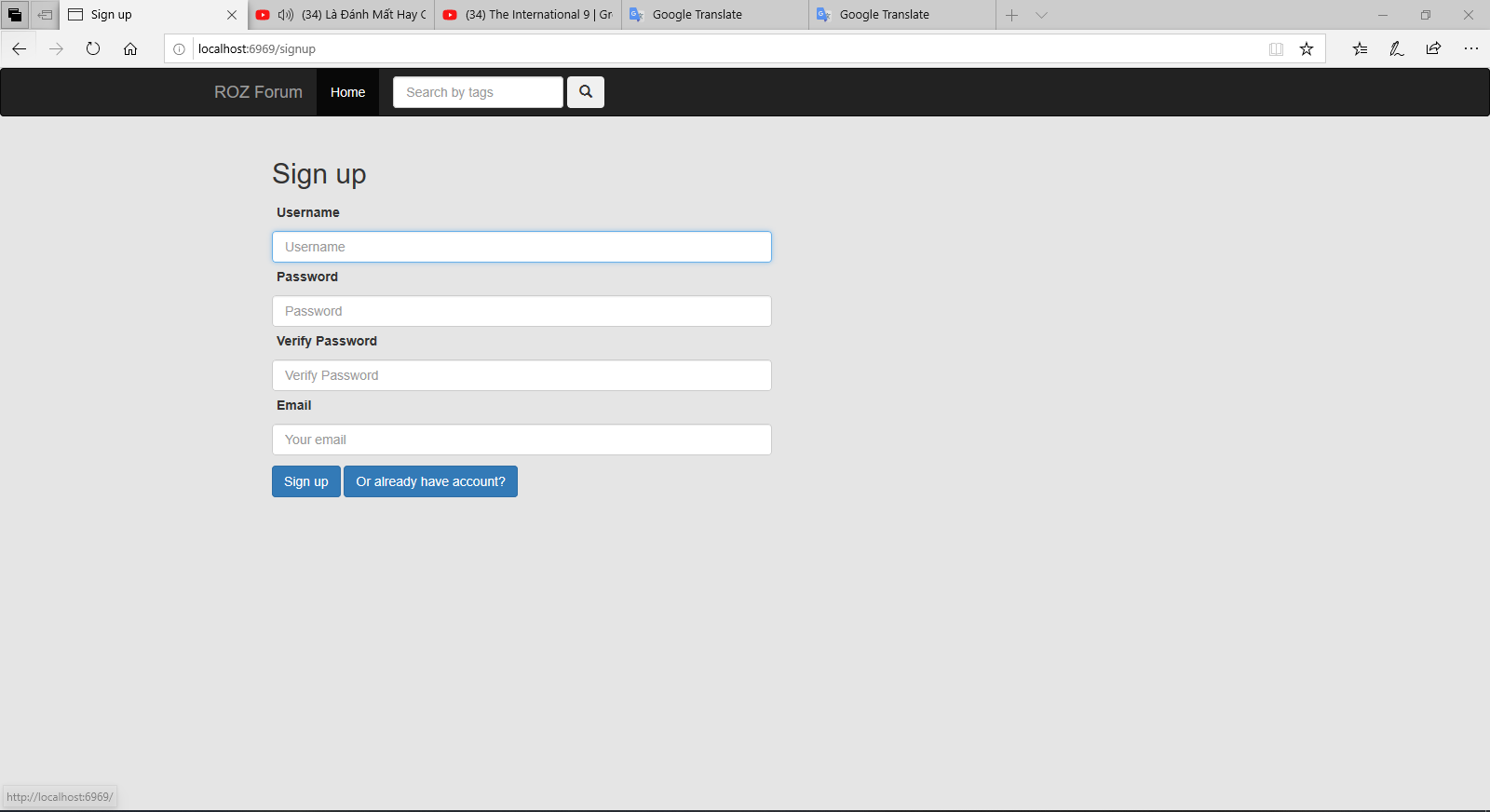


* Это HTML-тег «form». С помощью HTTP-метода «POST» для отправки данных пользователя на сервер.
* Я обновил "onsubmit" на этой "форме". Так что этот брaузер заметит, если пользователь напишет неправильно.

Kогда пользователь пишет пароль не правильно :

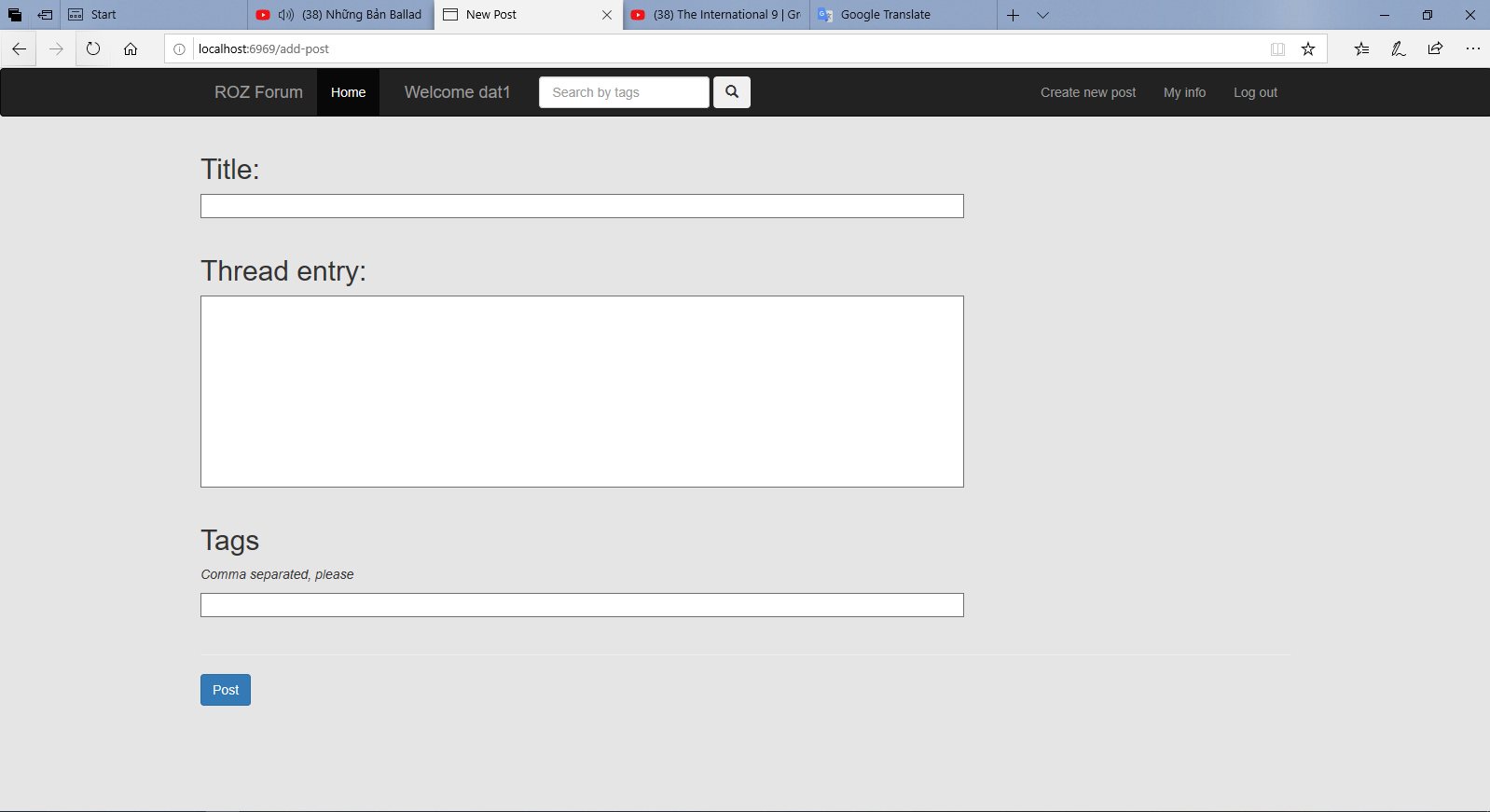


3: Cтраница для регистрации пользователя



* Это похоже на страницу для входа пользователя
* Я использовал "Regex” выражение, чтобы применить требование. Какой пользователь должен следовать за ними, чтобы зарегистрироваться

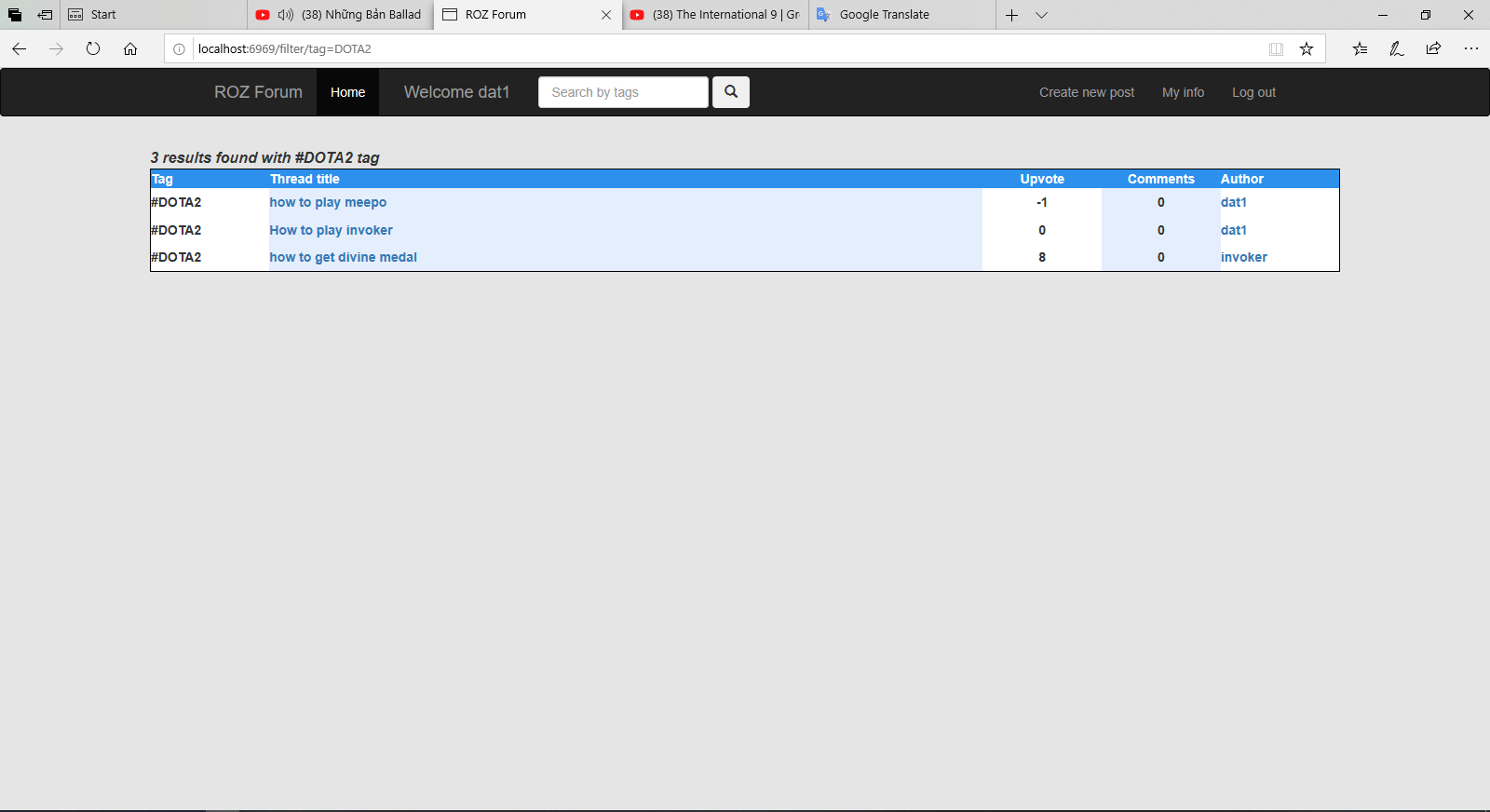
4: Cтраница для пользователя написать новую тему



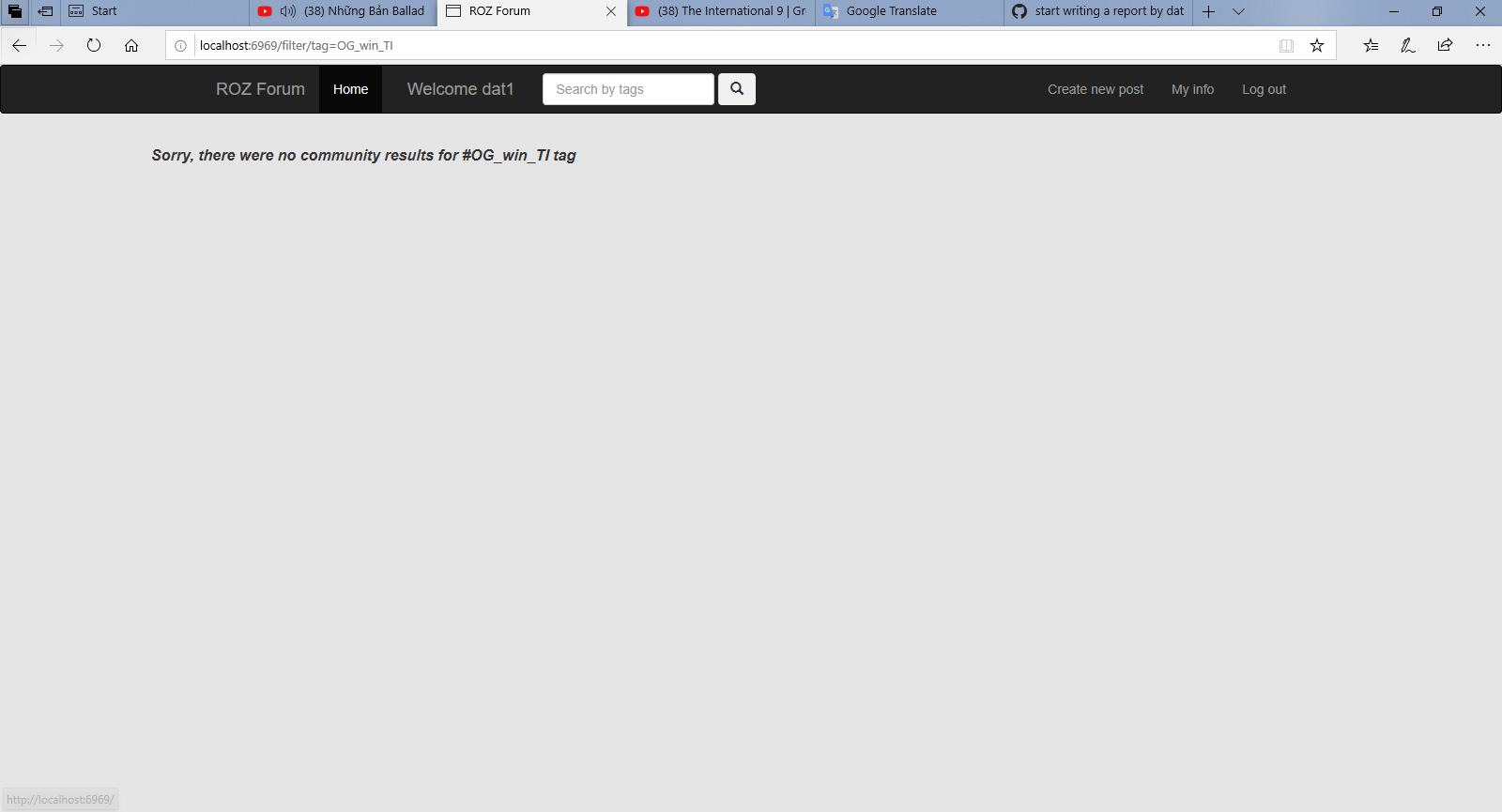
* Это HTML-тег «form». С помощью HTTP-метода «POST» для отправки данных темы на сервер.

5: Cтраница, показывающая все темы, которые пользователь ищет

Если сервер найдет какой-либо тем, который совпадает с пользователем хочет найти :

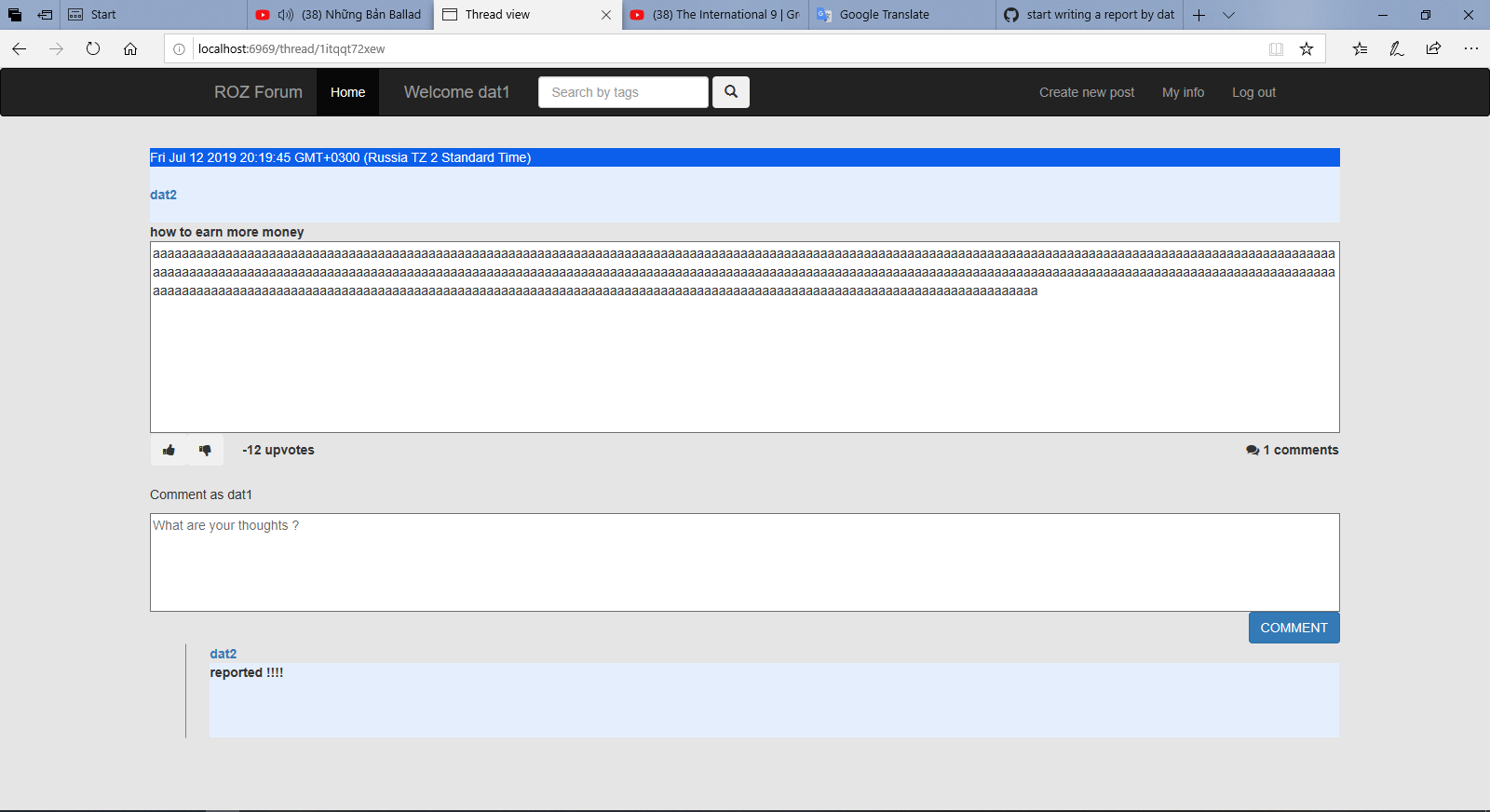


В противном случае:



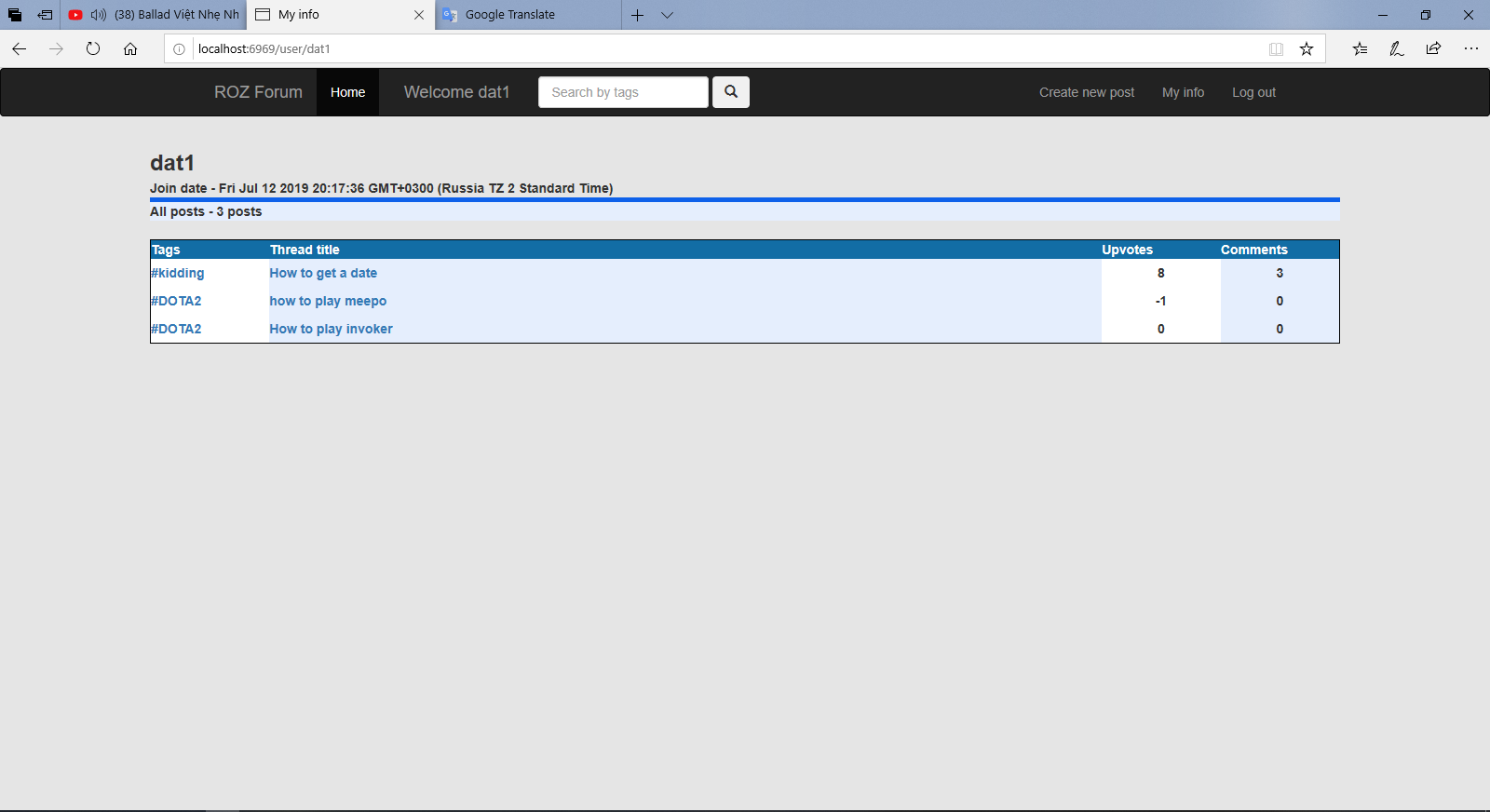
* Когда пользователь пишет тег, который он хочет найти в строке поиска (с энак «#» или нет, не имеет значения). Сервер будет искать в базе данных тем и находить все темы с этим тегом.

6: Cтраница для показа темы



* Эта страница показывает
  + время, когда автор пишет эту тему
  + автор, пользователь, который написал эту тему
  + название и содержание этой темы
  + кнопка нравится / не нравится, количество «upvote», количество комментариев
  + место для написания комментария к этой теме (показывать только когда пользователь в сети)
  + все комментарии, которые написал другой пользователь
* Когда пользователь нажимает на темы, который он хочет видеть на главной странице, сервер просматривает базу данных и показывает точно темы.
* Кнопка «нравится / не нравится» - это HTML-метод «POST» для увеличения / уменьшения атрибута «upvote» в данном базe.
* Каждая база данных тем имеет свою базу данных для комментариев. Публикация комментария - это HTML-метод «POST», он отправляет серверу информацию об этом комментарии, затем сервер найдет эту тему в базе данных и добавит комментарий в эту базу данных.

7: Страница для отображения информации о пользователе



* Когда пользователь нажимает на имя пользователя, которое он хочет видеть, сервер просматривает базу данных пользователя, чтобы найти этого пользователя для показа (имя пользователя на сервере только одно, при регистрации на сервере не разрешается повторять имя пользователя)

II: Back-end и база данных

1: База данных описания: