

System for managing gym
operations through a
distributed system

GYM MANAGEMENT SYSTEM

Functional Requirements



Admin: create, modify, and delete client profiles, personal trainers profiles and courses

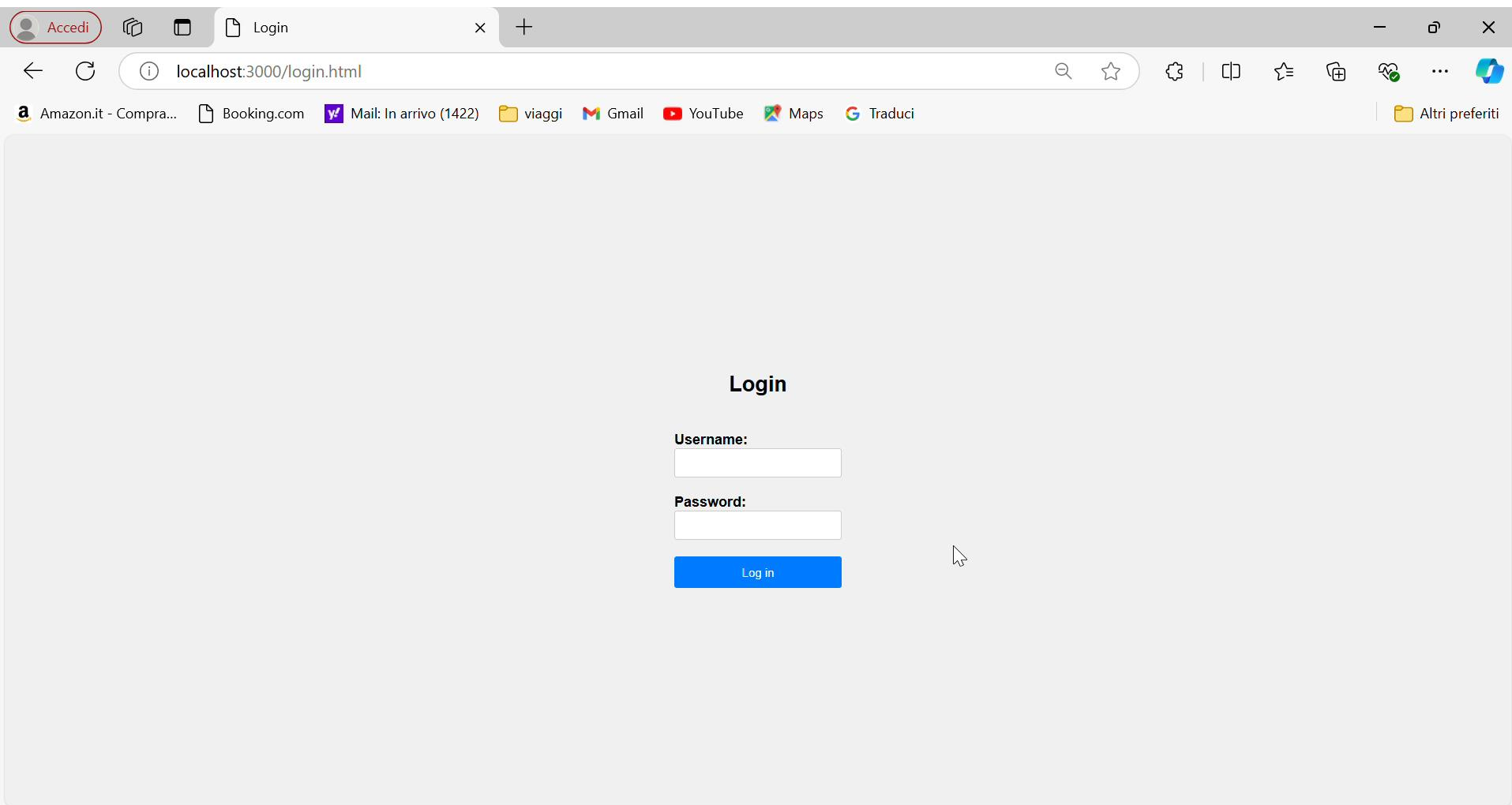


Personal trainers: view their profiles, courses, sessions and client profiles



Clients: view their profiles and bookings. Enroll/unenroll in courses and book/unbook private sessions

Admin access and creation of trainers, clients, courses



Accedi

localhost:3000/login.html

Amazon.it - Compra... Booking.com Mail: In arrivo (1422) viaggi Gmail YouTube Maps Traduci Altri preferiti

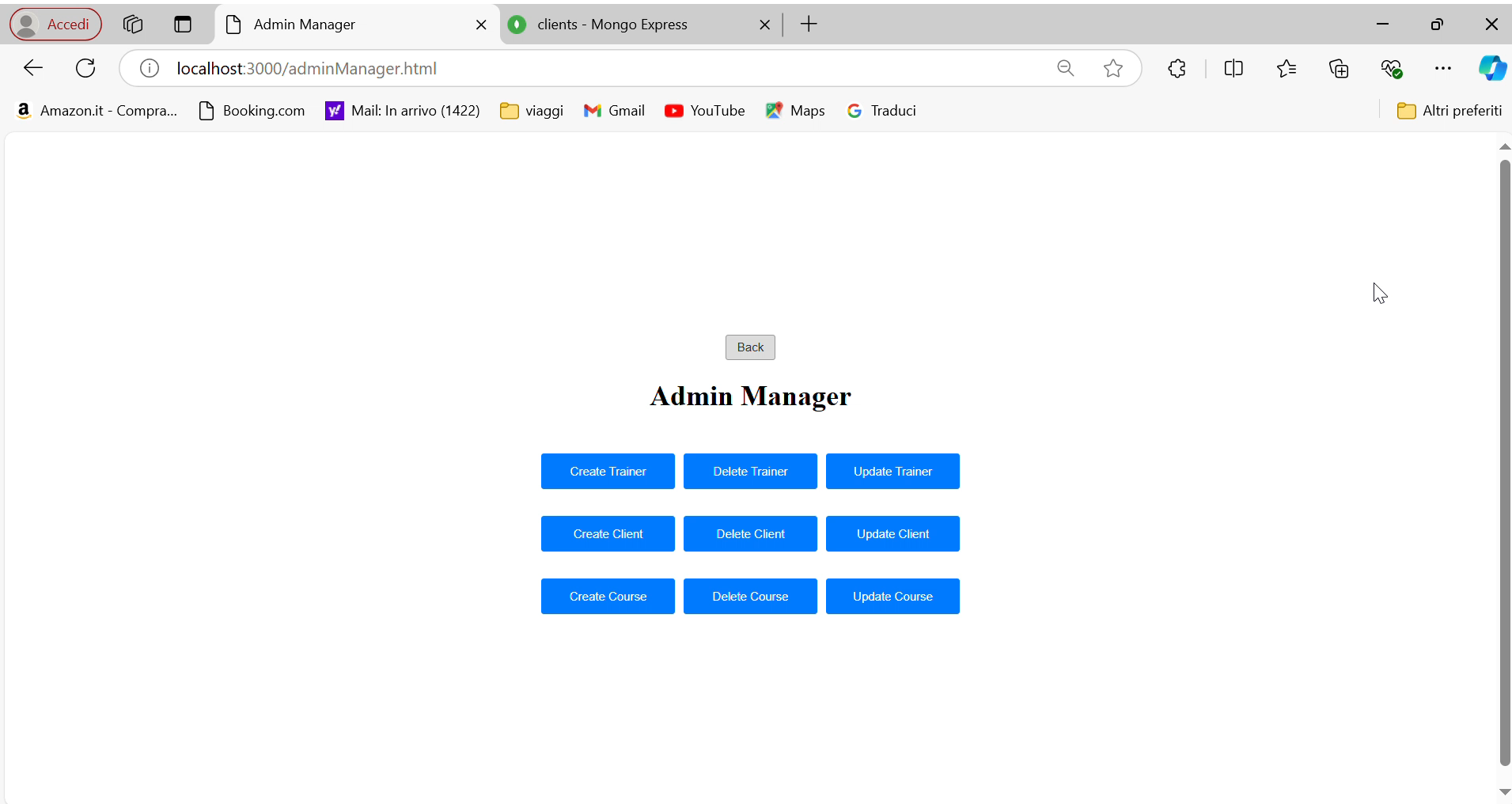
Login

Username:

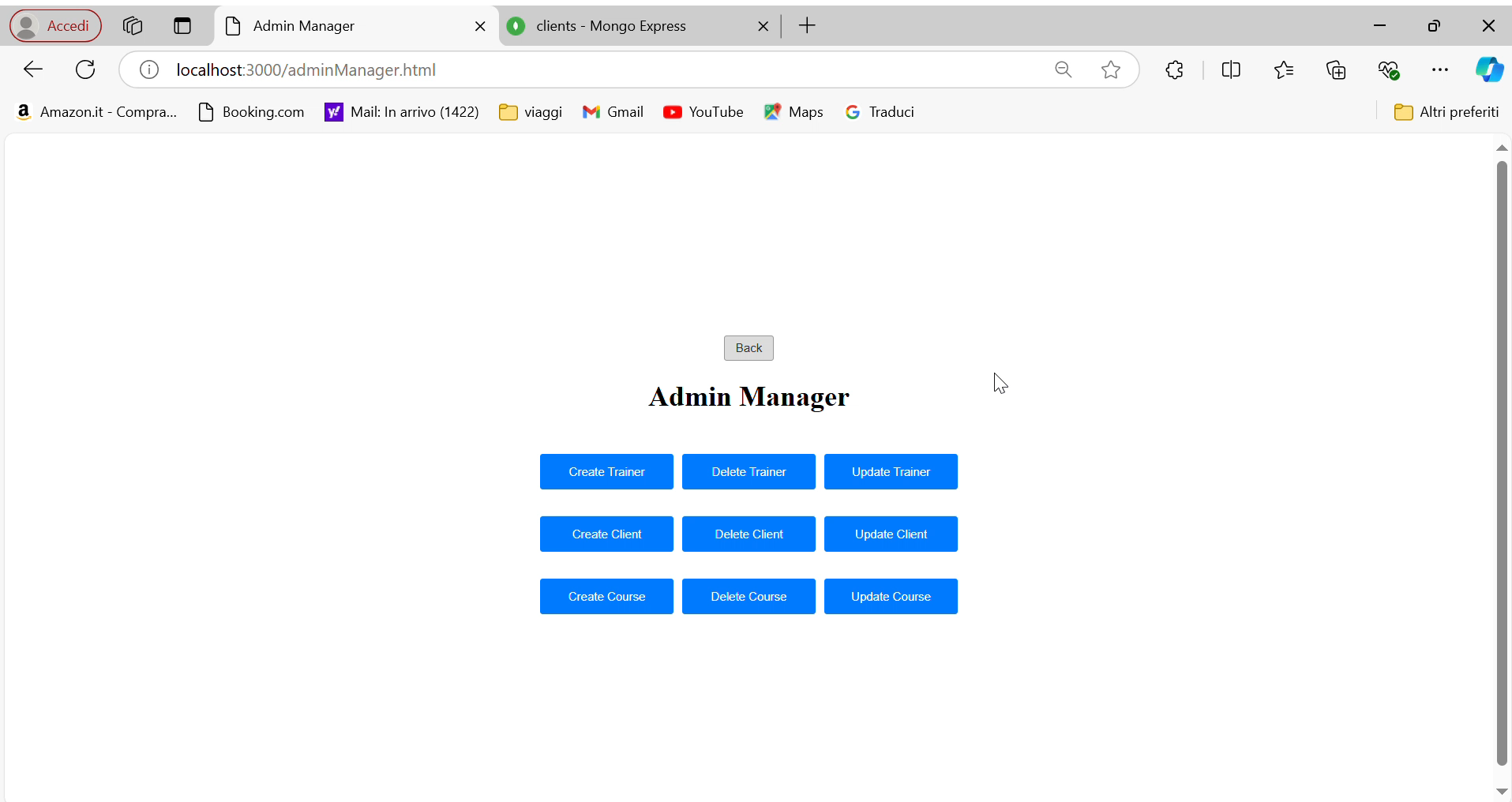
Password:

Log in

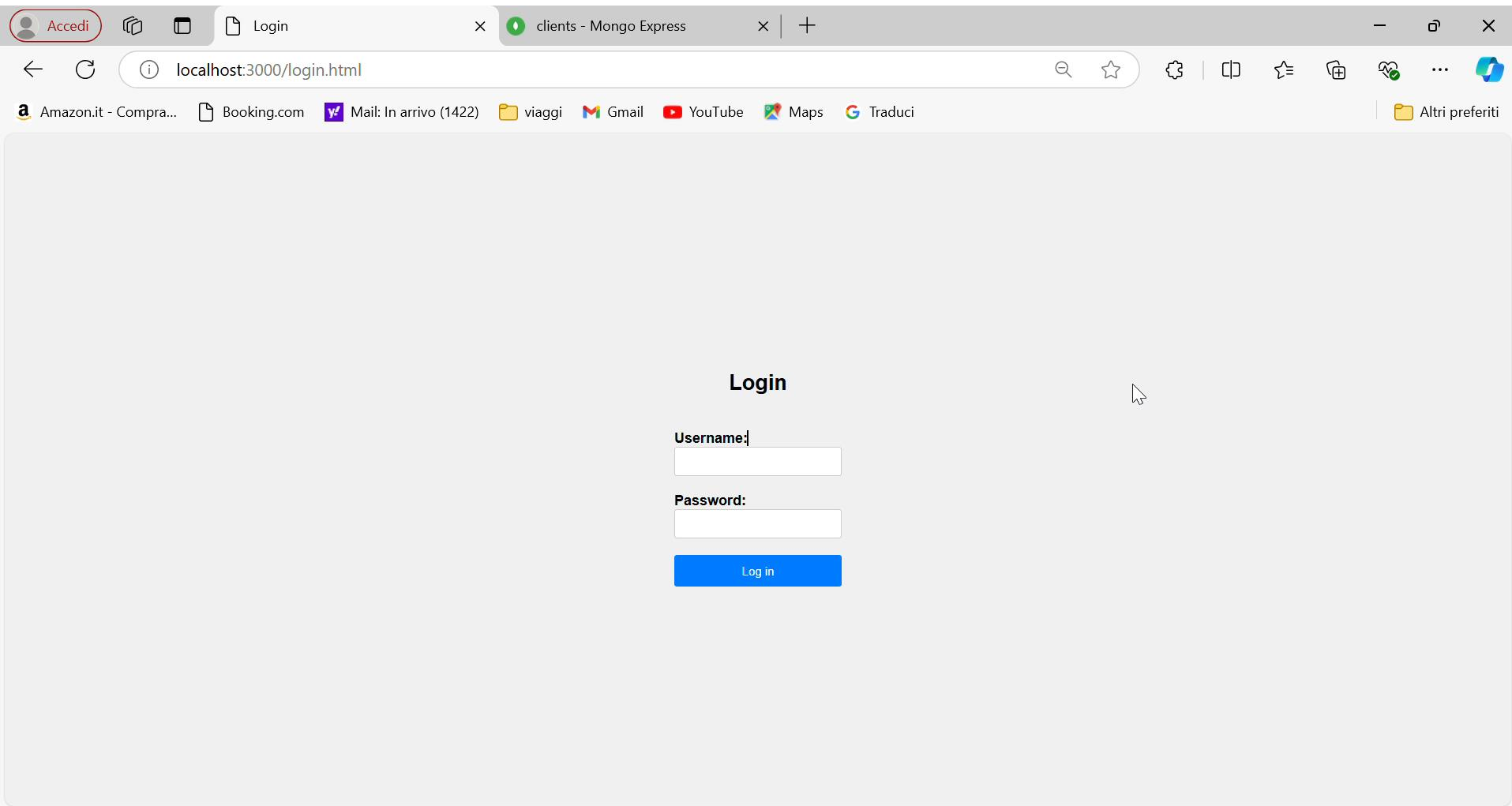
Update of trainers, clients e courses by the admin



Deletion of courses, clients and trainers by the admin



View client's personal area and courses and private sessions booking



Accedi

Login

clients - Mongo Express

localhost:3000/login.html

Amazon.it - Compra... Booking.com Mail: In arrivo (1422) viaggi Gmail YouTube Maps Traduci Altri preferiti

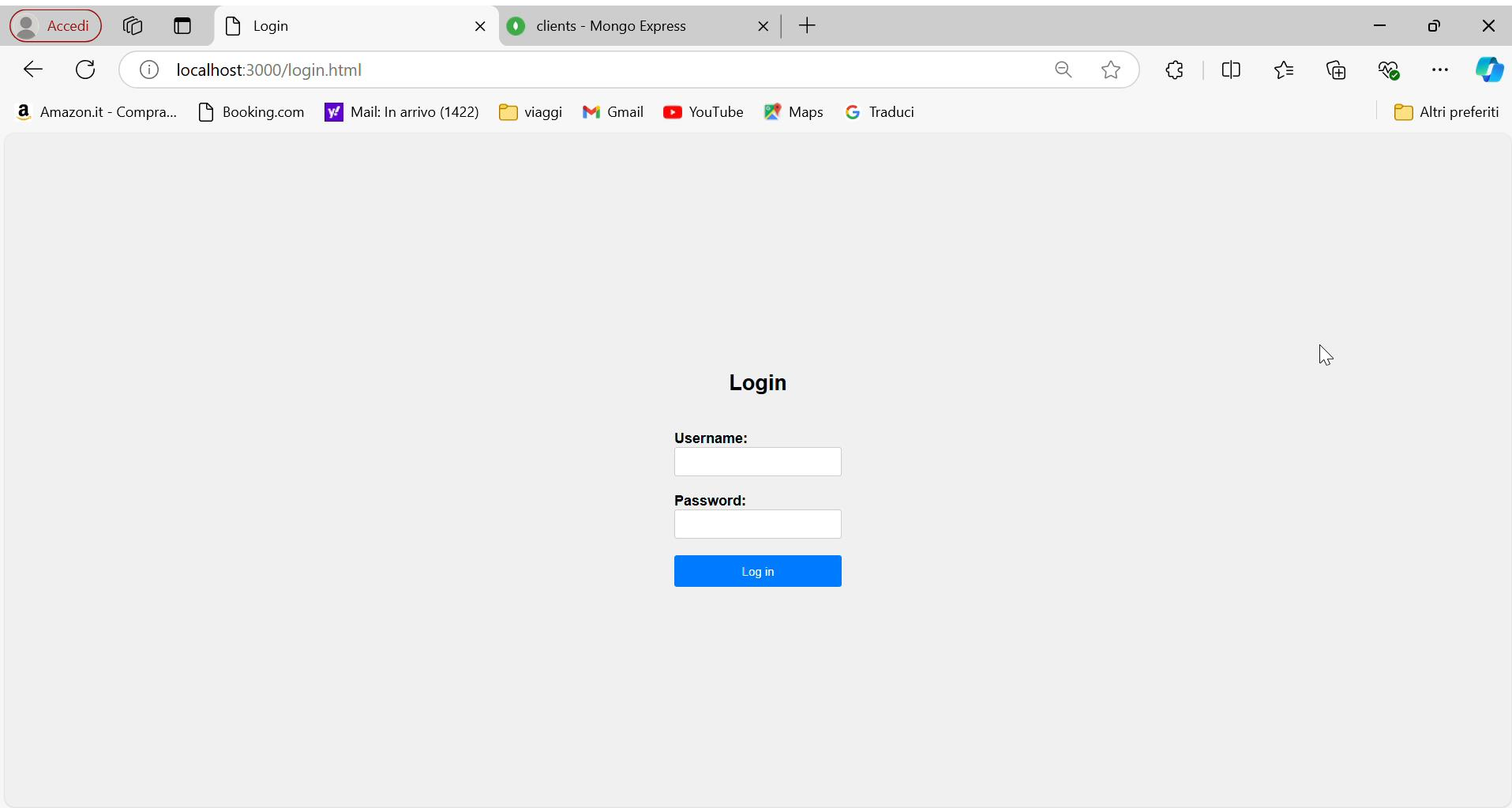
Login

Username:

Password:

Log in

View trainer's personal area



The screenshot shows a web browser window with two tabs: 'Login' and 'clients - Mongo Express'. The address bar displays 'localhost:3000/login.html'. The browser's toolbar includes navigation buttons, a search bar, and various utility icons. Below the toolbar, a row of bookmarks is visible, including 'Amazon.it - Compra...', 'Booking.com', 'Mail: In arrivo (1422)', 'viaggi', 'Gmail', 'YouTube', 'Maps', and 'Traduci'. The main content area of the browser is a light gray rectangle containing a login form. The form is titled 'Login' and consists of two input fields labeled 'Username:' and 'Password:', followed by a blue 'Log in' button. A mouse cursor is positioned to the right of the form.

Accedi

clients - Mongo Express

localhost:3000/login.html

Amazon.it - Compra... Booking.com Mail: In arrivo (1422) viaggi Gmail YouTube Maps Traduci Altri preferiti

Login

Username:

Password:

Log in

Non- functional Requirements



Concurrency control



Fault tolerance and
reliability

System Architecture:

Three-tier architecture



Frontend: HTML, CSS, javascript



Backend: Node.js, Express.js



Persistence layer: MongoDB

Database: MongoDB



1. Scalability



2. Concurrency
and Consistency



3. High availability

Mongoose



Formal definition of the document structure in the database



Specify fields, data types and constraints to ensure a consistent database



Manages data creation, update and read operations in a simplified manner

Containerization: Docker



Processes isolation



Resource management



Scalability



High availability



Fault tolerance



Distributed environment

Docker in the gym management system



Replica set with
MongoDB: high
availability and fault
tolerance



MongoExpress: web
interface for database
management



Application replicas:
high availability and
load balancing