

EVERYTHING FITNESS

"A year from now...
... you may wish...
You had started today."



DEVELOPED BY



Vlada Lukic
Java Developer
Phone: 065/3320-080
Email: vld.lukic@gmail.com



Marko Medik
Java Developer
Phone: 064/258-8261
Email: markomedik@yahoo.com



Sasa Milovanovic
JavaScript Developer
Phone: 065/449-9432
Email: sasa.milovanovic0606@gmail.com

Contents

Main Menu..... 3

Overview 3

Login..... 3

Register 4

Workout Generator 4

My Workout..... 5

 My Plans:..... 5

 Personal Details: 6

DATABASE AND DESCRIPTION 7

API CREATED 8

Main Menu

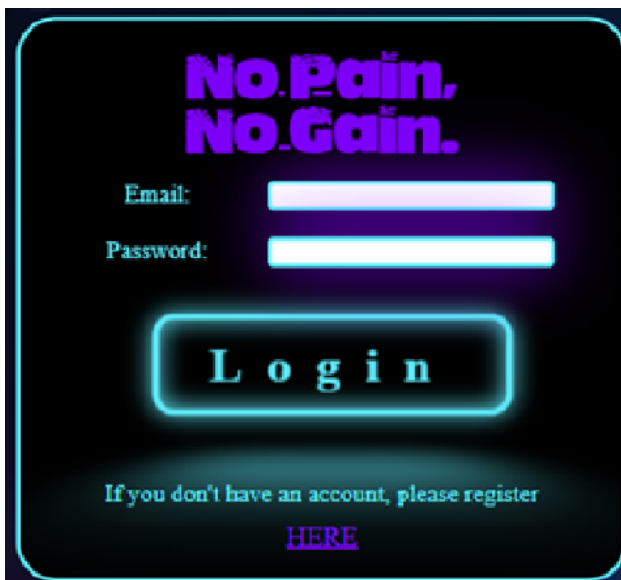
- Login,
- My Workout,
- Contact,
- About,
- BMI Calculator,
- BMR Calculator,
- Workout Generator

Overview

- When a visitor comes to the website, he will be able to access some of the features, such as **BMI** and **BMR Calculator**, **About** and **Contact** page.
- Visitors will be able to check our Workout plans in **Workout Generator** and read what there is to know about the specific plan, but they will not be able to generate any plans until they are logged in.
- **My Workout** section will only be available to logged in users. And this section will show generated plans and user details.
- Application architecture
 - Frontend: Vue.js
 - Database: MySQL
 - Backend: MARS

Login

On login, if user does not have the account, he will be able to register by clicking on the 'HERE' button that is in scope of Login form. By clicking Login, we execute Login API. API is executed in the auth.js Pinia store, where some of the information from the response is stored for future use in the rest of the application.



The login form features a dark background with vibrant neon green text and borders. At the top, the slogan "No Pain, No Gain." is displayed in a bold, stylized font. Below this, the labels "Email:" and "Password:" are positioned to the left of two white input fields. A large, glowing neon green button with the text "Login" is centered below the input fields. At the bottom, a line of text reads "If you don't have an account, please register" followed by a neon green link labeled "HERE".

```
async login(user) {
  const generalStore = useGeneralStore();
  try {

    const response = await axios.post(`${generalStore.API_URL}user/login`, user);

    const userRole = response.data.userRole;
    const userId = response.data.userInfo.user_id;
    const pictureId = response.data.userInfo.picture_id;
    const bodyId = response.data.userInfo.body_measurement_id;

    this.setAuthentication(true);
    this.setUserId(userId);
    this.setPictureId(pictureId);
    this.setBodyId(bodyId);

    return userRole;
  } catch (error) {
    throw error.response.data.msg;
  }
},
```

Register

The register consists of basic information about the user with several required fields marked with *. All these fields can be edited later in the Personal Details tab.

REGISTRATION

User Information

First Name *

Last Name *

Email *

Password *

Dob (yyyy-mm-dd)

Profile Image: *

Body Measurements

Gender *

Height

Weight

Fitness Level

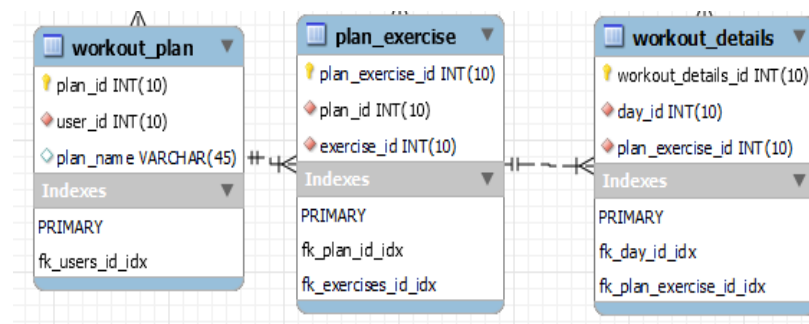
Fitness Goal

Fitness Goal *

* Required field

After the user is successfully registered and logged in, he will be able to use the website functionality in full. Firstly, user will be able to go to Workout Generator page and select one of the predefined workout plans.

Workout Generator



There is a list of Workout plans with plan descriptions and tips for properly executing the exercises. Once the plan is selected, by clicking. Generate Workout, an API will be run, and the weekly plan will be generated in the database (assigned to the user) and random exercises for selected plan will be generated and assigned to the plan. The user later will be able to navigate to My Workout section, select generated plan and see the exercises list performed by days.

☐ Push/Pull/Legs Split

☒ Full Body Workout

Full Body Workout

Ideal for: Beginners and busy bees with time constraints.

Why? This routine targets all major muscle groups, enhancing overall strength and functional fitness. Perfect for those just starting, typically requiring 3 days a week, with each session lasting 45-60 minutes.

Pro Tip: Remember, choosing the appropriate weight for each exercise is crucial. It should be heavy enough to reach the designated reps in a set, leaving 1-2 reps in the tank. This ensures optimal muscle engagement across 3-4 sets, with the last set challenging you to barely lift those last two reps.

☐ Upper/Lower Split

☐ Legs/Back/Chest/Arms Split

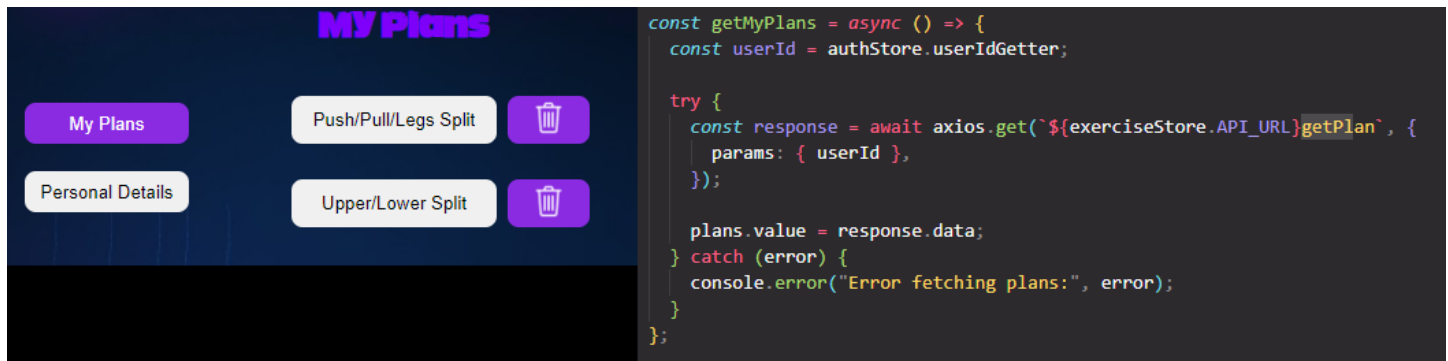
☐ Bro-Split

My Workout

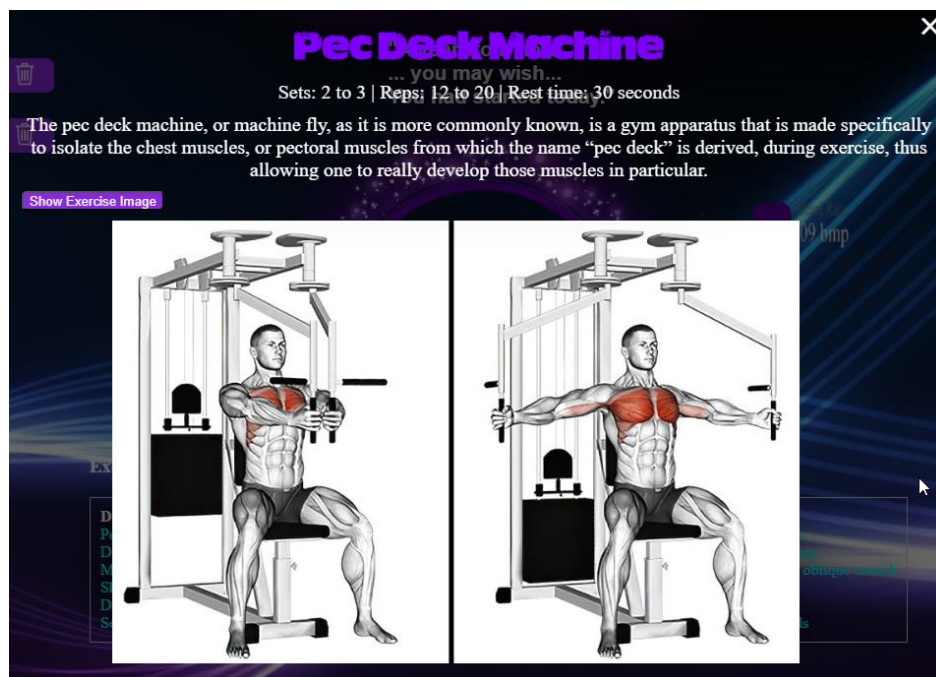
In my workout sections, there are two tabs available, My Plans and Personal Details.

My Plans:

Showing all generated plans, and once selected, will show the exercises per day generated for the user. Also, workout plans can be deleted from the list (and database).



By selecting a Workout plan, a new component is started that shows the list of exercises per day. Every exercise is clickable, and it shows a description of the exercise, how many sets and reps are needed to be performed with rest time, and if needed, showing image on how the exercise is performed.



Personal Details:

In the Personal Details tab, the user can see basic information typed in during registration process. From this tab, he can edit his personal information, or change password if desired.

The image displays three sequential screenshots of a web application's 'Personal Details' page. Each screenshot features a dark blue header with the title 'Personal Details' in purple. Below the header is a navigation bar with 'My Plans' and 'Personal Details' buttons. A circular profile picture of a man is shown. The main content area lists user details: Name: Marko Medik, Email: test@test.com, Date of Birth: 1987-01-01, Gender: Male, Height: 188 cm, Weight: 95 kg, Fitness Level: Advanced, and Fitness Goal: Endurance. The first screenshot shows the 'Edit' button. The second screenshot shows the form with input fields for editing: First Name (Marko), Last Name (Medik), Email (test@test.com), Date of Birth (1987-01-01), Gender (Male), Height (188), Weight (95), Fitness Level (Advanced), and Fitness Goal (Endurance). The third screenshot shows the form with password change fields: New Password, Confirm Password, and buttons for Confirm and Cancel.

The user details and image are called via two separate APIs, and show on the form:

```
const getUserDetails = async () => {
  const userId = authStore.userIdGetter;

  try {
    const userDetailsResponse = await axios.get(
      `${generalStore.API_URL}/user/userDetails?userId=${userId}`
    );

    firstName.value = userDetailsResponse.data.user[0].users_first_name;
    lastName.value = userDetailsResponse.data.user[0].users_last_name;
    email.value = userDetailsResponse.data.user[0].users_email;
    dob.value = userDetailsResponse.data.user[0].users_date_of_birth;

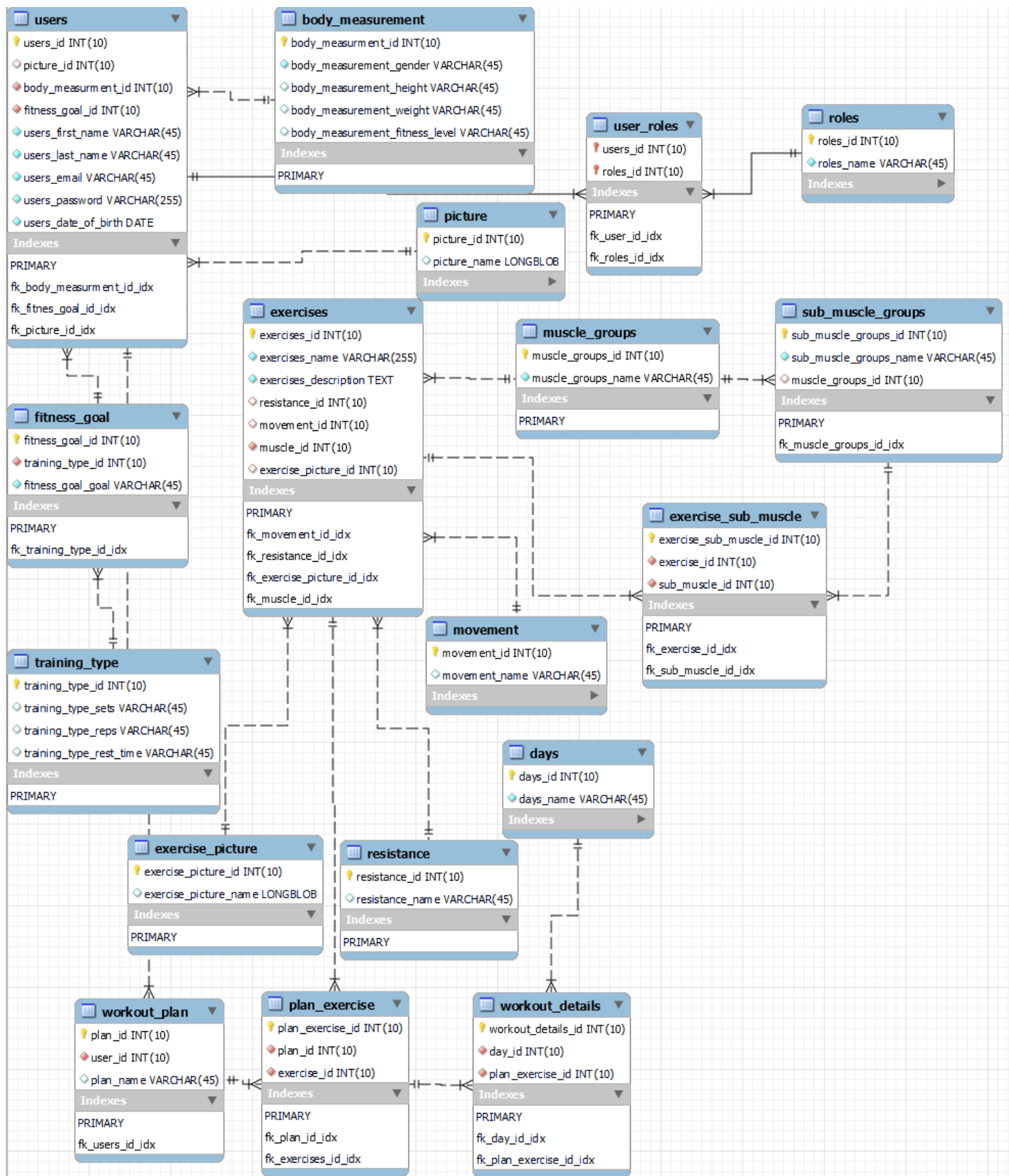
    gender.value = userDetailsResponse.data.user[0].body_measurement_gender;
    height.value = userDetailsResponse.data.user[0].body_measurement_height;
    weight.value = userDetailsResponse.data.user[0].body_measurement_weight;
    fitnessLevel.value =
      userDetailsResponse.data.user[0].body_measurement_fitness_level;
    fitnessGoal.value = userDetailsResponse.data.user[0].fitness_goal_id;
    userDetails.value = userDetailsResponse.data.user[0];

    userDetailsLoaded.value = true;
  } catch (error) {
    console.error("Error fetching user details:", error);
  }
};

const getUserImage = async () => {
  const pictureId = authStore.pictureIdGetter;
  try {
    const timestamp = Date.now();
    const userImageResponse = await axios.get(
      `http://228k123.e2.mars-hosting.com/user/getUserImage?pictureId=${pictureId}&timestamp=${timestamp}`,
      { responseType: "arraybuffer" }
    );

    const blob = new Blob([userImageResponse.data], { type: "image/jpeg" });
    const imageUrl = URL.createObjectURL(blob);
    userImageUrl.value = imageUrl;
  } catch (error) {
    console.error("Error fetching user image:", error);
  }
};
```

DATABASE AND DESCRIPTION



APIs

