



SL Fitness

Fundamentals of Script Programming: Final Project

András Mumm 72175404

Brian Lam 72175445



Table of Contents



Overview

Quick overview of SL Fitness's functionalities

01

02

Live demonstration

Live demonstration of SL Fitness

Technology stack

Frameworks used to build SL Fitness

03

04

Obstacles

Challenges faced while developing SL Fitness



01

Site overview

Quick overview of SL
Fitness's functionalities

Features



User Registration

Create your own account
to display your own fitness
statistics



Performance Tracker

Record your performance
for multiple exercises and
track them through graphs



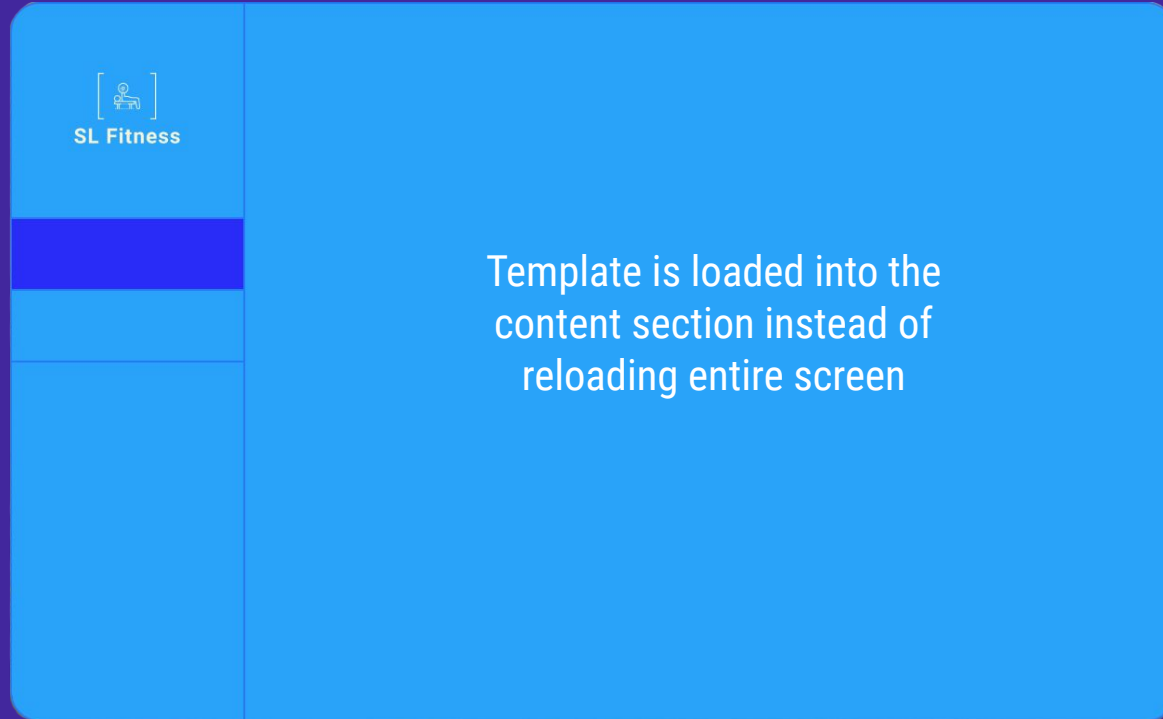
BMI Tracker

Record your weight and
height to calculate your BMI
and track it through a graph

Dashboard components

Nav bar

Content



Registration

Login to view your dashboard

Username*

Password*

Login

[Create an account](#)

Create an account

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Register

If you already have an account, [login](#) instead.

Performance tracker



Enter the name of the workout and an initial value of what you are currently capable of:

Workout Name:

Start Value:

BMI tracker

Great to see you test123,

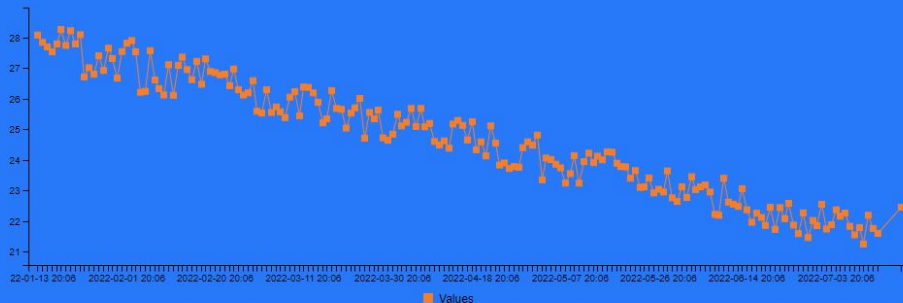
Your last recorded weight: 76.0kg

Your last recorded height: 184.0cm

Your last calculated BMI: 22.45

Add today's
measurements

BMI statistics



Enter your current height and weight

Height(cm):

Weight(kg):

Submit



02

Live Demo

Live demonstration of
SL Fitness

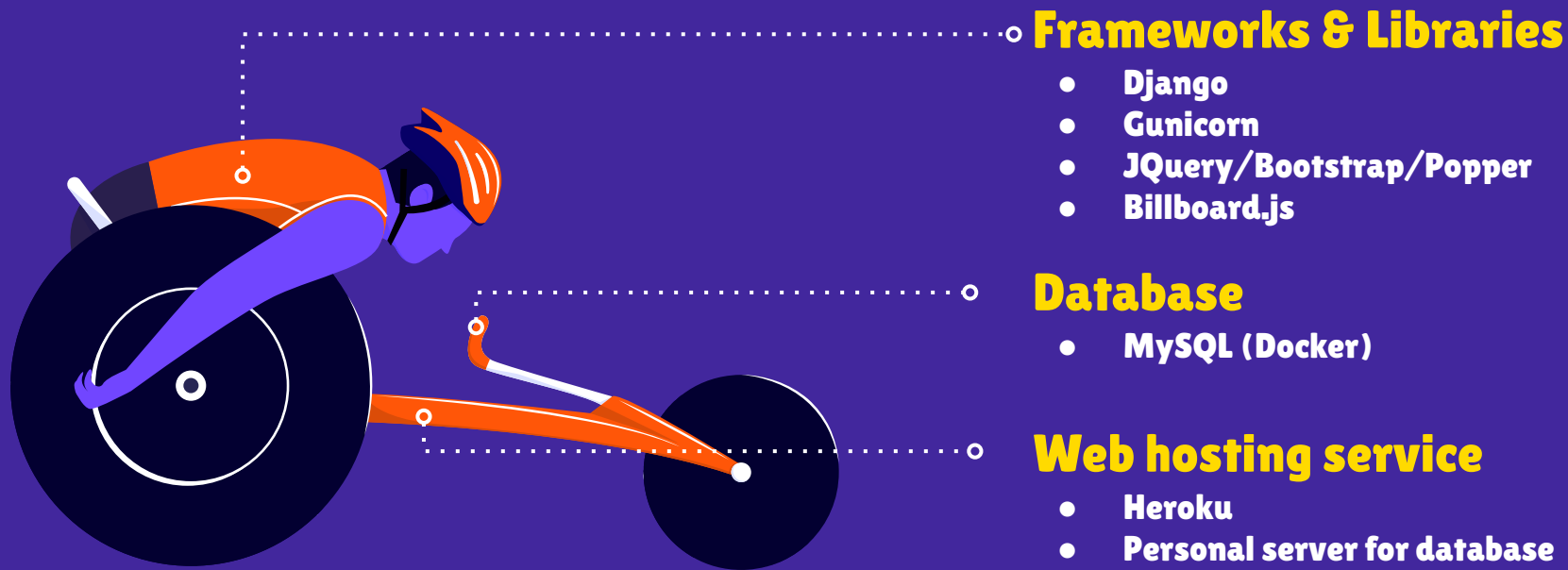
03

Technology stack

Frameworks used to build
SL Fitness



Technology stack



Django Features Used

- Django forms
 - Create forms for user to input fitness related data
 - Built in AuthenticationForm and UserCreationForm for user registration functionality
- Django template
 - Usage of template language (blocks, if statements ...)to change parts of the dashboard instead of reloading the entire screen
- API (Django REST Framework)
 - Models
 - Custom Serializers
 - Custom Viewsets
 - Full functional REST API.
This allows for the creation of a desktop client or a mobile app using the same backend.



04

Challenges

Challenges faced while
developing SL Fitness

Challenges

- Learning the Django Framework
 - Understanding the REST framework to create the API
 - The template feature allows for block based website creation and removes the need for dynamic frontend programming, but requires a more systematic approach on how to implement the backend.
- Figuring out where to implement Asynchronous Programming
 - Our program has strictly sequential code execution requirements:
 - On a POST (update) request, can only send back an updated render of a view if the update has been fully processed and the database is updated.
 - On a GET request, we need to preprocess the data in sequential order to allow for performant client side charting (no need for sorting, as the database entries are already sorted)
 - Therefore adding asynchronous codes would have over-complicated and slowed down the backend.
 - We have decided against adding asynchronous processing.

Thanks for listening!