SAPIENZA
UNIVERSITÀ DI ROMA

LA SAPIENZA- UNIVERSITY OF ROME
FACULTY OF INFORMATION ENGINEERING, INFORMATICS, AND STATISTICS

Cloud Computing
a.y. 2021/2022

PROJECT PROPOSAL

MediCareWizard

Martina Milazzo - 2026374
milazzo.2026374@studenti.uniroma1.it

Clizia Giorgia Manganaro - 2017897
manganaro.2017897@studenti.uniroma1.it

Edoardo Di Martino - 1821427
dimartino.1821427@studenti.uniroma1.it

Leonardo Plini - 2000543
plini.2000543@studenti.uniroma1.it

1. Idea

Nowadays, modern booking systems are capable of making our lives easier by facilitating getting an appointment and speeding up the time needed for it.
With the idea in mind to avoid the potential long queues we can face when needing to book a medical visit, we decided to create MediCareWizard: a system for booking medical appointments at the patient's fingertips.
MediCareWizard aids the patients in need of a specialist's examination or a medical service by creating an easy to use, fast and immediate system to satisfy their needs, with the aim of guaranteeing a quality service and short waiting times within our diagnostic medical center.

2. Implementation

In order to implement our project we would like to use the AWS Cloud Platform, since we think it is a valid service to create a secure and scalable solution.

To allow users to choose, book, delete or modify an appointment, we must provide a web interface: the idea is to use some technologies such as html, css, js, php and Bootstrap to create a responsive website.
All the features will be implemented inside **Amazon Web Services** as microservices: users will interact with the website, using indirectly AWS microservices to book, delete and modify appointments.
The website will be hosted inside an **Amazon CloudFront** instance, that will be linked (with the Amazon API Gateway) to the microservices that we will use to access the Database instance.
The microservice that we would like to use is **AWS Lambda** in order to use **Amazon Aurora** and **Amazon DynamoDB**.

We want a fully scalable solution, so the idea is to split all the services that we will implement, and implement these separately, so that they can be edited or scaled on demand.

Users will be able to sign in and login in, search for the specific clinic, query to choose the specific type of medical visit that they would like to do, specify the day and the hour, and to edit the booking afterward.

2.1. Technologies

Here follows two separate lists for the technologies that we will use, the first one for the frontend and the second one for the backend.

Frontend:

- HTML5
- CSS
- JavaScript
- Bootstrap

Backend:

- Amazon S3 Bucket
- Amazon CloudFront
- Amazon CloudWatch
- AWS Lambda
- AWS Databases (Aurora and DynamoDB)

3. Test

In a second moment, we will want to control the delays caused by multiple and simultaneous virtual user actions.
To validate and test our microservices we will use the built-in function Amazon CloudWatch. In order to test our performance we will need some testing toolkits, such as Artillery.io or Apache JMeter, that will simulate different levels of workload.