Contents

1 Introduction

The Progressive Web App (PWA) allows users to create, post and search for events. Users are able to take selfies through the use of WebRTC or select a photo from a folder to be posted, where each post can receive likes and comments which is implemented with SocketIO. Each user can follow other users and vice versa. Attended and interested events are stored and displayed for each user. Web workers are used to allow offline usage with IndexedDB used to store data locally. MongoDB is used to store data which can only be retrieved online. The search function (via location) was implemented with the use of Google API, allowing selected location to be displayed on the map.

2 Diagrams

Diagram 1 displays the flow of each web page and the transmission and retrieval of data.

3 Interface to Insert and Search Data via Forms

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

4 Interface to Search Data via Map

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

5 PWA – Caching of the App Template Using a Web Worker

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

6 PWA: Caching Data Using IndexedDB

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

7 NodeJS Server Including Non-Blocking Organisation of Multiple Dedicated Servers

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

8 MongoDB

- **Challenges**:
  - Loading of initial data to populate the database.
  - The retrieval and storage of images in the form of bits.
- **Solution**:
  - Initial population of database was resolved with the use of promises.
  - Image retrieval and storage was implemented using *base*64 encoding.
- **Requirements**:
  - Data stored in MongoDB is stored online. Data is not stored locally and cannot be retrieved offline. When user is online, data is retrieved from the database and displayed on the PWA.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

9 Quality of the Web Solution

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

10 Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

11 Division of Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

12 Extra Information

- **Initial population of MongoDB**: run 'node run initdb' to drop database and repopulate database with default data.