Contents

[Introduction 1](#_Toc117762027)

[Overview 1](#_Toc117762028)

[Background 1](#_Toc117762029)

[Core Functions 1](#_Toc117762030)

[Advanced Functions 1](#_Toc117762031)

[Data Protection 1](#_Toc117762032)

[Implementation 2](#_Toc117762033)

[User Interface 2](#_Toc117762034)

[Technology Stack 2](#_Toc117762035)

[Data Organisation 2](#_Toc117762036)

[Hosting 2](#_Toc117762037)

[Services 2](#_Toc117762038)

[Scalability 2](#_Toc117762039)

[Tracking and Statistics 2](#_Toc117762040)

[Conclusion 2](#_Toc117762041)

[References 2](#_Toc117762042)

# Introduction

Do towards end of development.

Describe problem

State scenario with details of selected cultural festival.

A local film festival has put out a tender for a new website to serve their needs. They require a website to advertise the festival as well as functioning as a hub, directing users to the appropriate places to be able to find out information about locations, what’s on, and being signposted to where to book tickets.

Background of scenario and client requirements

The festival is based in the Glasgow area and uses existing cinema facilities such as the GFT theatre, the Grosvenor, the Everyman cinema as well as certain Cineworld and Odeon screens. The client is looking for a website which informs its users of what’s on, where and when, a section with background information on the festival and previous contributors, images or photos of previous festivals, and the ability to search the available events. The initial prototype will show the basic functionality as a proof of concept with further advanced functionality included where time is available or indicated in the report.

Initially, to keep things simple and cost effective for a small group of largely voluntary employees, 3rd party APIs will be used for ticket booking, e.g., EventBrite. The user would click a link on the festival site and be automatically redirected to EventBrite login page and the rest of the transaction would be handled by EventBrite.

There will be an admin login function, so events can be added and updated or removed by festival staff. This will require a simple username and password – salted and hashed before being stored in the database – and role types – to allow a specific admin user to add or update but not delete, for example.

Highlight any key points.

Legal – GDPR/Data Protection compliance statement.

Using 3rd party apps to process personal user data, payments.

# Overview

## Background

Data sources app will use – public, private, licensed APIs

Mapping service such as Google Maps, HERE maps, or Mapbox.

Database/API to serve images and information to display to users

EventBrite or similar to handle event scheduling, booking, and ticketing.

User inputs

A contact form for more information about the festival.

Login feature to track history.

## Review of competitive products looking at common core features and advanced features

<https://glasgowfilm.org/glasgow-film-festival>

<https://www.edfilmfest.org.uk/visiting>

<https://whatson.bfi.org.uk/lff/Online/default.asp?menu_id=D90AD2E1-12B8-4B8D-97A2-DB9A5A4BC4A4>

For a local film festival, we reviewed 3 different festival websites, the Glasgow film festival, Edinburgh International film festival and the BFI London film festival with a view to comparing the common features across all three then assessing what would be regarded as the core features to make a functioning website and what would be an advanced feature to provide more value to users of the website. This will help us in our development of a prototype proof of concept and guide any future developments.

Across the 3 websites the common features are background/history of the festival with an About us section, A means to donate to support the festival, a section containing the latest news, A newsletter signup form.

Each of the websites has a means of logging in or signing up, a ticket purchasing section or signposting to a third-party site or other means of booking tickets.

Review of competitive products looking at common core features and advanced features

Development time and cost

Does the proposed app meet the brief

Purpose of site. Project scope.

## Core Functions

Determine requirements of the app –

In an initial launch of the app there will be certain functionalities required to allow the clients to get the best use of the website at launch.

Display schedule – showings with locations and film information.

History of event - articles

Photo gallery

Key people involved

What’s on

Search events

Login functionality for an administrator to add festival events.

Booking and Ticketing – redirection to 3rd party.

Design (logo, branding, colour scheme) - Professional and modern layout, topic suitable, ease of use/navigation, suitable for those with disabilities, e.g., using a screen reader, etc.

Responsiveness – The app should be responsive for different device screen sizes.

Accessibility - Allow users to navigate the web site using touch or a mouse or keyboard or to search the site for information or to view a site map, since the site may be used by people of all ages and abilities.

Site must be usable by as wide an audience as possible,

User login required? Profile/account area – display personalised information, age-restricted content?

What data is needed

Static images/injected images/graphics - in-house advertising (events/films) and external advertising (e.g., AdSense)

Video clips (trailers)

Navigation menu – must be simple and easy to use: or must be intuitive and clear.

Calendar section – showing events and availablitity of tickets. - pull data from 3rd party app?

Simple form to contact us – collect name, email address, select dropdown reason – volunteer, sponsor, marketing, other – type thing? Auto sends email

Payment/ticketing - 3rd party app such as EventBrite.

Database, to include -

* Database information for Films, Locations, Events, and Admin Users.
  + Films would contain for example the title, the tagline, a short summary blurb, the lead cast and director, a static advert image, a short trailer video.
  + Locations would contain for example the name of the Theatre, the address, the phone number, the website, the contact email address, a short description about the theatre, a list of onsite facilities.
  + Admin Users would contain username, salted hashed password, roletype.
  + Events would contain the film ID from Films table, dates it’s showing, times it’s showing, location id from the Locations table (which theatre(s) is showing the film), possibly a link or unique id which leads to a specific page on e.g. EventBrite.
* If Events booking is being handled by a third-party, it doesn’t need to show availability or any further detail, as the user can click the link to go to EventBrite and see the up-to-date information there.

Profile information

User login required? Profile/account area – display personalised information, age-restricted content? Third party app access – eg forwarding to EventBrite or similar.

## Advanced Functions

Advertise for volunteers

Apply for sponsors

Apply to take part?

Add a user login area? Like a members-only login.

Simple form to contact us – collect name, email address, select dropdown reason – volunteer, sponsor, marketing, other – type thing? Auto sends email – use 3rd party handler for that?

Static/injected notices filling on-page banner spaces – sell space to local advertisers.

## Data Protection

Security and how protected.

3rd party apps to process eg payments so no sensitive data held by actual festival company. GDPR still currently in force.

# Implementation

## User Interface

## Technology Stack

## Data Organisation

# Hosting

## Services

To deploy a project, we will require a hosting service to deploy the prototype onto to make the project available to show the client. There is an option to build and run a client’s own server to deploy the project and serve customers but that would involve added cost and technical specialists to run. Using a cloud-based solution is a more cost-effective solution to project deployment in which costs start at a free tier for small projects rising as more resources are used. This allows a project to grow naturally as well as any related costs rising alongside.

A hosting service such as Heroku would be a suitable solution as they provide low-cost solutions to deploy an application. Heroku is a Platform as a Service which is based on a managed container system and was created to provide an application focused approach (Heroku, n.d.). Heroku can use either their own CLI or GitHub integration to allow for quick uploading and updating of any code changes to which are made to an application this leads to a quick turnaround for any updates to be published to the production server once tested.

Heroku also provides many tools to manage the troubleshooting of issues one of these tools, Heroku OpEx can alert a developer to issues that arise quickly allowing for a quick response in identifying and solving problems. Using Heroku OpEx a developer has access to Logs to receive data regarding any errors or events, statistics such as the memory and data usage of a dyno or how quickly the dyno is serving HTTP requests. All information which can be used to inform the best way to use the platform. This dashboard also allows for Autoscaling of the dynos to quickly respond to increases in traffic to ensure the application does not struggle with unexpected traffic.

## Scalability

Any website or application which experiences any kind of significant growth will need to be able to scale to meet that demand. As this project is a data driven application it will be important to consider a scaling solution which considers the security and the integrity of the data which is being used. There are two main types of scaling to consider Vertical and Horizontal.

Vertical scaling is where a hosting service adds more computing power to the existing application to increase the computing power such as the CPU, RAM and DISK to deal with increasing workload. This, however, is restricted to the limits of the server which it is hosted on and as such there may be a risk of having downtime or a service outage if the demand reaches higher than can be currently coped with.

Horizontal scaling is where more machines are added to the resources available to the application and processes may be split among these extra machines to balance the user connections to the application and share the load.

Heroku uses a container system to run an application and initially this will be deployed on one web dyno (Heroku container) to serve the web app. As this project uses Node.JS which is JavaScript based technology and is therefore a single-threaded language, an API called Cluster is required to run multiple processes at the same time. This can be enabled in the CLI and will allow the app to be able to scale through the use of multiple dynos (Heroku, 2022). A developer can choose from the many solutions offered and can upgrade or downgrade a dyno quickly to be able to respond to the traffic an application is receiving.

This offers the flexibility of horizontal and vertical scaling and provides a good solution using the best method to serve an application. Setting up an application to run on multiple dynos also offers a level of protection against an outage as they are more likely to be run on different servers, meaning that if one goes down the application would still be available to users. This is one of the main benefits of using a cloud-based solution to serve a web application.

## Tracking and Statistics

# Conclusion

# References