**Portfolio Website**

Lavdim Ismaili – [isla21xr@student.ju.se](mailto:isla21xr@student.ju.se)

Web Development Fundamentals - Jönköping University - 2023

*In this template, created by Peter LARSSON-GREEN, all italic text should be removed and replaced with your own text (which should not be italic); the italic text is just a placeholder letting you know what to write there.*

*On the cover page, change to your own project name, your own name and your own JU email address.*

*You have a lot of freedom when it comes to writing this report. You do not have to use any parts of this template, but the report you write should in the end somehow (in a good way) provide the same information as indicated in this template. Most students trying to do it in their own way usually fail, so if you try that, be sure to know what you are doing!*

*Use proper sentences, paragraphs, lists, tables, figures, etc. The more figures you use, the less text you need to write, so use many of them.*

*It's amazing (scary) how many students that delete this text before they have read/understood it…*

Contents

[Introduction 0](#_Toc144993168)

[Method 1](#_Toc144993169)

[Architecture 1](#_Toc144993170)

[Database 1](#_Toc144993171)

[Graphical User Interface 1](#_Toc144993172)

[Web Application 2](#_Toc144993173)

[Results 3](#_Toc144993174)

[Discussion 3](#_Toc144993175)

# Introduction

*Introduce your project here. Write text that* ***indirectly*** *(look up what indirectly means if you don't know it) answers questions like:*

* *Why does the project exist?*
* *What is the project about?*
* *Who are involved in the project?*
* *What will the project result in?*
* *Who are interested in the outcome of the project?*
* *How will the outcome of the project be used?*
* *...*

The portfolio website project is a digital representation of my achievements and expertise. It provides a platform for me to showcase my work and skills, making it accessible to potential employers, clients, and collaborators. This digital identity is a valuable tool for networking, personal branding, and professional growth. It exists to bridge the gap between aspirations and accomplishments, offering a gateway to new opportunities and broader horizons.

*.*

# Method

## 2.1) Architecture

*En bild som visar skärmbild, diagram, text, design

Automatiskt genererad beskrivningGive an overview of the components the website consists of (web application, database, web browsers, end-users, etc.). Visualize this in a figure and show how the different components make use of each other.*

*After having read this chapter, the reader should have a broad (but shallow) understanding of the website's internal structure.* ***DO NOT*** *describe implementation details of the individual components, you have other chapters for that.*

* End-users (web browsers) send requests to the web application by entering the website's URL or clicking on links.
* The web server receives these requests and forwards them to the web application.
* The web application processes the requests, which may involve fetching data from the database.
* The web application generates responses, typically in the form of HTML, CSS, and JavaScript.
* The web server sends these responses back to the web browsers, which render the content for users to see.

## 2.2) Database

*Describe the database and the resources on the website in detail. What attributes do they consist of? How are they related? Where are they stored (in files on the hard-drive? In a relational database? Etc.). Visualize the resource in an ER diagram or similar, for example as the one found at* [*https://www.researchgate.net/figure/Database-Schema-for-storing-the-processed-version-of-BBC-News-dataset\_fig5\_339168671*](https://www.researchgate.net/figure/Database-Schema-for-storing-the-processed-version-of-BBC-News-dataset_fig5_339168671)*.*

These resources are typically stored in a relational database, which could be, for example, a MySQL or PostgreSQL database. Here's how the resources and attributes are related in a simplified Entity-Relationship (ER) diagram:



*After having read this chapter, the reader should understand how the data (the resources) on the website is stored. If the reader is a new programmer that should start working on the website, she should now know what she needs to do if she wants to change the resources or add more type of resources (e.g., know how to add a new table to the database with a relation to an existing table in the database, etc.).*

## 2.3) Graphical User Interface

*Describe the graphical user interface of the website. Initially this can simply be low-detailed pictures, as the ones shown in Figure 1 below (Note: the click at the top is on the "Contact" link, and not the "Home" link (optical illusion)).*

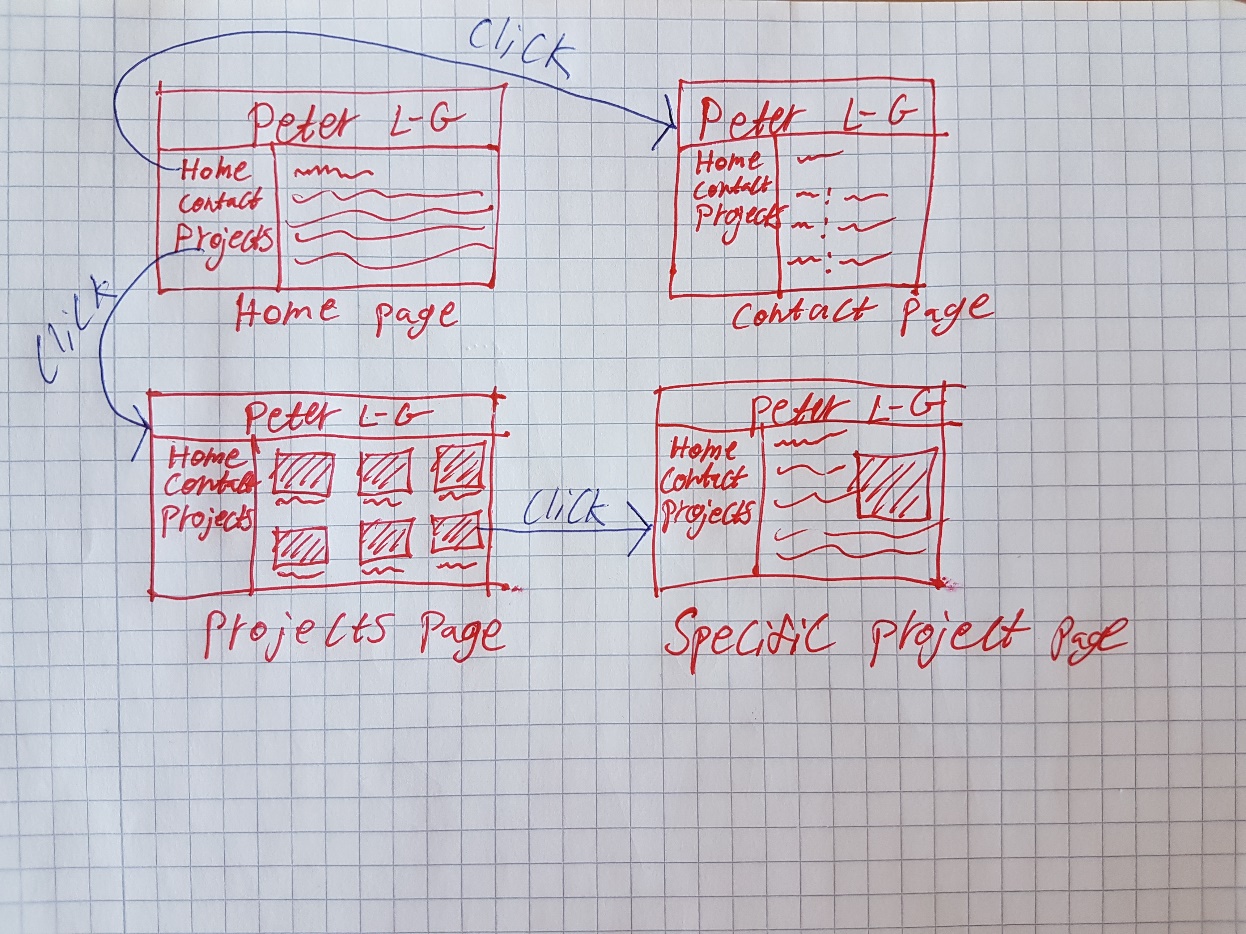


Figure 1, Low-detailed picture of a graphical user interface.

*When you have implemented the website, showing some screenshots of it here is a very good idea. Keep your low-detailed pictures even when you have real screenshots; they give a good/quick overview of the GUI.*

*After having read this chapter, the reader should understand how the end users will be able to accomplish their goals (use-cases from your use-case diagram) through the graphical user interface.*

Home, about, contact, login (4 pages)

*A screenshot of a computer

Description automatically generated*

## 2.4) Web Application

*Describe implementation details of the web application. Which language have you used? Which framework have you used? Which libraries/packages have you used, and for what purpose? Has all code been written in one file? Or have you somehow structured it in multiple files? Are you using some design patterns (e.g. MVC)? Are you using middlewares? Etc...*

*You do not necessarily need to show any code to describe the implementation, but if you feel that improves the quality of the report, feel free to do that.*

*Try to use many figures. They are much easier to read than a wall of text. Use text to explain details that can't easily be visualized in a figure. File structure, code structure (functions, classes, etc.) can be visualized using various types of UML diagrams, etc.*

*Security is a very important topic when it comes to web applications. Having a sub-chapter here that describes all security vulnerabilities you have thought of (e.g. using HTTPS (and why), hashing passwords (and why), XSS, cookies, session ids, ...) is probably a good idea. It's probably a good idea to have some more sub-chapters as well, but you can figure out them on your own.*

*After having read this chapter, the reader should have a very good understanding of how the web application has been implemented. If the reader is a programmer who should start working on the web application, she should now know where to start when she should implement new features to the web application.*

# Results

*Describe the results you obtained (you can include several screenshots of your final project). How did you ensure the security of your project?*

# Discussion

*Discuss about the overall project, what you have learned in terms of organization of the work, new software and technologies…*

# References

*Include your references following the APA recommendations (*[*https://guides.library.ju.se/apa-en*](https://guides.library.ju.se/apa-en)*) from Jönköping University Library. There are many available resources available to help you writing a good report on the library website in the “Search & Write” menu (*[*https://ju.se/library*](https://ju.se/library)*).*