Assignment 1

BurgerHub

PA1489

Group 19

Alem Zvirkic(alzv23), Amir Hindawi(amhi22), Theodore Reangpusri(thre21)

Introduction

For this assignment we followed the instructions published on canvas.

Everyone is going to create their own menu, with burgers, side dishes, drinks and they get to name it as they want. We are going to split up the work into three areas of responsibility, Theodore is responsible for interface and github management, Alem is responsible for functionality and Amir is responsible for database and docker. Everyone has the responsibility to test the project by creating automated tests.

Timeplan

Below is our timeplan for the whole project. More detailed weekly plans can be viewed in our trello.

Week	38	39	40	41	42
To be done	Assignment 1 done	Assignment 2 done	Assignment 3 done	Summarize text and diaries	Complent. Turn in the project

It is possible to follow our weekly progress by accessing our Trello in which we have created an active issue board.

Bestäm utvecklingsmiljö

Developer Environments: Visual Studio Code, Docker
Revision Control: Git, Github
Coding Languages: Javascript, SQL, HTML, CSS
Frameworks: Tailwind
Project management and communication: Trello, Discord

We chose these tools and technologies because we have prior experience working with them, which helps streamline our development process. **Visual Studio Code** offers a lightweight yet powerful development environment with numerous extensions, while **Docker** ensures consistent environments across machines. For version control, **Git** and **GitHub** allow for easy collaboration and tracking of code changes, which we've successfully used before. Our coding stack of **JavaScript, SQL, HTML, and CSS** is ideal for building a dynamic, data-driven web application, and **Tailwind CSS** helps us quickly create responsive, clean designs with minimal overhead. This combination allows us to work efficiently while leveraging industry-standard tools.

Bestäm git-server

We chose GitHub due to our previous experience with the platform and its widespread usage in the tech industry. Many major companies, including 90% of Fortune 100 businesses, rely on GitHub for collaboration and version control, which gives us a competitive advantage over others who might not be as familiar with Git. This familiarity ensures we can work efficiently from the start, leveraging GitHub's tools to manage our codebase effectively.

GitHub statistics

Skapa projektet

BurgerHub is an interactive web-based platform designed to enhance the burger ordering experience. It allows customers to browse and order a variety of burgers and side dishes directly from their devices. One of the standout features is the extensive customization options, where users can create their perfect burger by selecting from a wide range of ingredients, including various types of meats, vegetables, and other toppings. The site is intuitive and user-friendly, offering a seamless way to place and track orders, making it the go-to destination for burger enthusiasts looking for a personalized dining experience.

Katalog planering

We have created a github repository with all members and used git commands in the terminal in order to push and pull changes. More can be read in our diaries.

Git commands and reflections

- **Vilka git-kommandon använder ni? Vad gör de?**
- **Vilka svårigheter stöter ni på? Hur löser ni dem?**

Our diaries can be found in the *Reflektioner* folder which can be accessed via our github. The most common git commands that we use are:

Command	What it does	Comments
git fetch	Downloads objects and references from another repository without merging the changes.	We do this sometimes use this when we are unsure if we have all branches locally
git pull	Fetches from the remote repository and merges into the current branch.	We always start our session with pulling code from the main branch in order to not risk any potential merge conflicts.
git status	Shows the working tree status, indicating changes that are staged or unstaged	We try to use git status as much as possible.
git add	Stages changes for the next commit	We usually use git add . in order to add all files
git commit	Records changes to the repository with a descriptive message.	git commit -m “text” so send commit messages. We are keeping the messages short but informative
git stash	Temporarily saves your modified/staged files without committing them.	Useful for switching branches without committing.
git switch <branch name>	Switches branches in your working directory.	
git switch -c <branch name>	Creates a new branch and switches to it immediately.	

Initially we did use git init in order to actually use the git commands and push to github. So far the only difficulties that we have faced are estimating how much time and effort that is needed for this project, how to work between branches without merge conflicts and initial setup for the actual codebase. By overcoming these by reading documentation and searching online for answers. We had a discussion regarding our prior experiences, strengths and weaknesses before creating the timeplan in order to estimate hours.