

# TMA 01

---

BSC (HONOURS) COMPUTING AND IT

Matthew Mason: C6122243

BSC (HONOURS) COMPUTING AND IT | FRIDAY, MAY 17, 2024

# Table of Contents

PREPARATION AND PLANNING .....	2
WORKING TITLE .....	2
AIMS AND OBJECTIVES .....	2
DESCRIPTION AND SCOPE .....	2
Background .....	2
The Problem .....	2
Why is it a problem? .....	2
Benefits .....	2
Solution .....	3
ICT Aspects .....	3
Experience .....	3
Specialism .....	3
Project Output .....	4
Success Criteria .....	4
TASKS AND SUBTASKS .....	4
Part 1: General .....	4
Part 2: Responsive Web Application Design .....	4
Part 3: Building a Database .....	5
Part 4: Developing APIs .....	5
LIFECYCLE MODEL AND SCHEDULE .....	6
Justifying lifecycle choice .....	6
RESOURCES, SKILLS AND METHODS .....	7
Key Resources .....	7
Key Skills .....	7
Risk Assessment and Management .....	8
PROJECT WORK .....	11
LITERATURE SEARCH AND REVIEW .....	11
ADDITIONAL WORK .....	13
REVIEW AND REFLECTION .....	14
FEEDBACK/GUIDANCE REQUEST .....	14
References .....	15

# PREPARATION AND PLANNING

## WORKING TITLE

Developing a bespoke, responsive web application booking system for a chain of boardgame venues

## AIMS AND OBJECTIVES

- To develop a bespoke and responsive web application booking system using HTML, CSS, JavaScript and other frameworks, that's accessible from a range of devices
- Develop an appropriate project idea by iterating, refining and adjusting its scope and boundaries.
- Design and build a database.
- Develop a set of APIs to access and manipulate data.
- Extend and use knowledge of computing and IT subjects specific to the project.
- Successfully and independently manage a project from beginning to end.
- Identify, gather, utilise, analyse, and evaluate resources relevant to the project.

## DESCRIPTION AND SCOPE

### Background

The current reservation process at a chain of family run board game cafes is inefficient, ineffective and plagued by a multitude of mismanagement complications and difficulties. To reserve a table at any one of the venues, guests must call the specific venue and speak to a staff member who will record the booking via a reservation sheet, checking availability against future booking that can be found attached to the staff notice board. Customers can also email the business whereupon the administrator will coordinate with management at each venue to confirm and adjust the booking as necessary.

### The Problem

Multiple lines of communication and paper-based system facilitates a lack of cohesion and communication between parties which has cause a number of problems including, but not limited to:

- Miscommunication or reservation details between staff
- Misplaced forms leading to unknown bookings and the loss of customer details.
- Recording of incorrect information.
- Double bookings.
- Incorrect cancelled booking removal.

### Why is it a problem?

These problems lead to the business being unreliable in the eyes of potential customers and this inability to proficiently provide a consistent, reliable and trustworthy service causes lost business, revenue, trust, and projects a bad reputation online and through word of mouth. To alleviate these issues, the business has decided to invest in a booking system that will hope to provide a number of benefits:

### Benefits

- A reduction in training costs by assigning management to only senior staff member as staff turnover can be relatively high in the service industry.
- Bookings can be made outside normal operating hours increasing business.

- By having the ability to make an online booking in conjunction with a well-planned SEO optimisation strategy, it can improve click through rate and increase business via the booking system.
- Prevent double bookings and lost customer data.
- Reduces recording of incorrect information.
- Provides historical data of reservations if a problem arises.
- Reduces lost bookings and cancelled bookings.
- Can provide multi-user management permission levels.
- Automated acknowledgements and reminders.
- Virtual Events

### Solution

A potential solution is the development of a bespoke, responsive web application booking system that can expand functionality along with the business' needs. The system would allow customers to make reservations at any venue using a variety of devices (PC, tablet, mobile), select specific tables, time, dates, request additional features and reserve specific games, manage their booking using a unique reference number, hire private rooms, request games masters and beginner sessions and being a dialogue with management for booking special events. It would also allow management to access the system using assigned credentials, set up new users and manage customer bookings.

### ICT Aspects

There are a number of potential ICT aspects to this solution which will need to be considered:

- A variety of tools will need to be used in the development of a responsive web application such as Visual Studio Code, Chrome Dev Tools, GitHub, GitBash CLI, etc.
- The application of previous level 3 modules TM352 and TM356.
- An extension of academic learning to build on level 3 skills including building and deploying an SQL type database, developing APIs, and connecting front and back-end systems.
- Implementing security measures to comply with GDPR and other legal requirements regarding the storage and transit of customer data.
- A range of writing, spreadsheet and organisational software for project management, report writing, note taking and scheduling.

### Experience

I have a good understanding of the problem through my own experience of working as a host at a chain restaurant. I was in charge of managing bookings, seating arrangements, and setting customers' expectations. We used an extremely similar style of tracking reservations providing first-hand experience of how problematic and unorganised this system can be. Only parties larger than 6 guests were permitted to make bookings due to table size and other business factors (mainly getting groups in and out as fast as possible) and were recorded in an A4 day-to-view notebook along with daily tasks. However, these rules were often waived and occasionally bookings were forgotten or not communicated amongst me, and other staff cause frustration, especially during the more chaotic peak operating hours.

### Specialism

Developing the system as a web application provides utilising my knowledge of at least two of the previous modules I have studied; TM352 and TM356. It also allows the opportunity for independent learning and extending my skillset through a number of challenges such as database design and

development, API development, connecting front and back-end systems, project management and critical evaluation techniques.

### Project Output

The final project output will be a demonstration of a working booking system, comprising a variety of functionality for customers and management and the projects management code. It will also produce a report detailing the project and answering questions such as if it was successful, if and how it achieved its outcomes and goals, what I did well and what I could do better. An attached list of appendices will include project diaries, scheduling breakdowns, resources, literature references and a thorough analysis and evaluation of the project as a whole.

### Success Criteria

- Add, Modify and cancel a reservation to the database (as a user).
- Set up and assign credentials to employees to access the system.
- Add, Modify and Cancel a reservation (as an admin user).
- Export database records by specified time period (as an admin user).

## TASKS AND SUBTASKS

### Part 1: General

1. Managing work schedule: Incorporate specific periods at the beginning of the project and after each assignment submission to review tasks and subtasks adjusting them as necessary.
2. Evaluate literature: Use study and library resources to help evaluate if literature is appropriate for my project.
3. Refine project: Use tutor feedback, study resources and my evaluation of work and progress to make required changes if necessary.
  - 3.1. Refine goals, aims, objectives and success criteria.
  - 3.2. Refine boundaries and scope of project.
4. Identify key skills and activities: List all skills I feel I currently have, all skills that might be required for the project, how I will acquire them, all the resources I will need, specialist equipment, etc.
5. Risk assessment of project: determine the risks of my project, the impact they might have and how they will be mitigated.
6. Manage and organise reviews with tutor.
7. Reviewed, evaluated, chosen and modified lifecycle to projects context.
8. Implemented a strategy for developing skills: Planning study sessions into the schedule.
9. Research what systems are already available: Use the internet to search for similar systems by looking at different establishments websites (cafes, restaurants, bars, etc).
10. Set up Git and GitHub.
11. Evaluate work and progress: Use study resource "Evaluating your work" to review progress and help plan the next steps of the project.
12. Legal, Social, Ethical and Professional Issues: Use study resources to investigate and manage any necessary measures.

### Part 2: Responsive Web Application Design

Worked on from beginning as it will involve a number of repetitions and redesigns based on feedback and goals.

1. Wireframes: By using information readily available about the business, knowledge of interaction design (design principals) and research on a variety of other booking systems I will produce initial sketches of potential layouts for the following:

- 1.1. Desktop, tablet and mobile site wireframe.
- 1.2. Admin management console wireframe.
2. Prototyping – design low-fidelity prototypes: After gaining feedback and eliciting requirements from potential users, I will evaluate and analyse the data and produce new designs of the system using PowerPoint.
3. Prototyping – HTML, CSS, JavaScript, Frameworks: After a number of data gathering sessions and evaluations, I will begin to create a high-fidelity prototype of the system using typical web development languages.
4. Implement solution: After final consultations, will evaluate and analyse feedback and begin final coding process.
5. Testing of implemented code: Will iterate the testing phase after implementation of redesigns.
6. Evaluate feedback – Discover requirements: Initial consultation with potential users of the system to gain an understanding of processes and functionality required, establishing use cases and user stories for designing prototypes.
7. Redesign: Sketching or prototype creation of new layouts and alternatives based on evaluation and analysis of feedback.
8. Usability and user experience goals: Use literature (Interaction Design – 4<sup>th</sup> ed) to decide on the importance and priority of different usability goals and user experience goals.
9. Literature search: Use the internet, the OU library resources specific to my study, and other mediums to find literature and material that is appropriate to my project and will help extend my knowledge and learning (Development of web application, incorporating accessibility, designing database schema, building APIs, securing data in transit).
10. Establish users, stakeholders and context: Use literature (Interaction Design – 4<sup>th</sup> ed) to understand who my users and stakeholders are, their capabilities (cognitive, social, emotional), cultural factors and the context in which the application will be used.

### Part 3: Building a Database

This should be started ASAP and will extend learning.

1. Research – How to build a relational database: The majority of the research and analysis phase will be using the Codecademy interactive tutorials to understand the process of building a database which will then be applicable to my project.
2. Designing a schema and relative databases as needed: Using knowledge from research, literature and tutorials will allow me to create the databases needed.
3. Development – creating the required tables: After initial designs have been completed for what is required, development can begin.
4. Testing – Using SQL to edit and modify data.
5. Evaluation and analysis of design.
6. Implementation – Applying the database to the project to meet its goals.

### Part 4: Developing APIs

Cannot be started until basic web app and database has been established.

1. Research – How to build an API: Similar to building a database, the knowledge gained to extend my learning will be gained through Codecademy and material acquired from literature searches.
2. Connect front and back-end systems: Once a basic website, database and API are created, I will begin to develop the code that connects all the pieces together.
  - 2.1. Coding – implement APIs.
3. Testing – use success criteria: There will be a number of testing iterations once code has been implemented to check for bugs, and errors that need to be fixed.



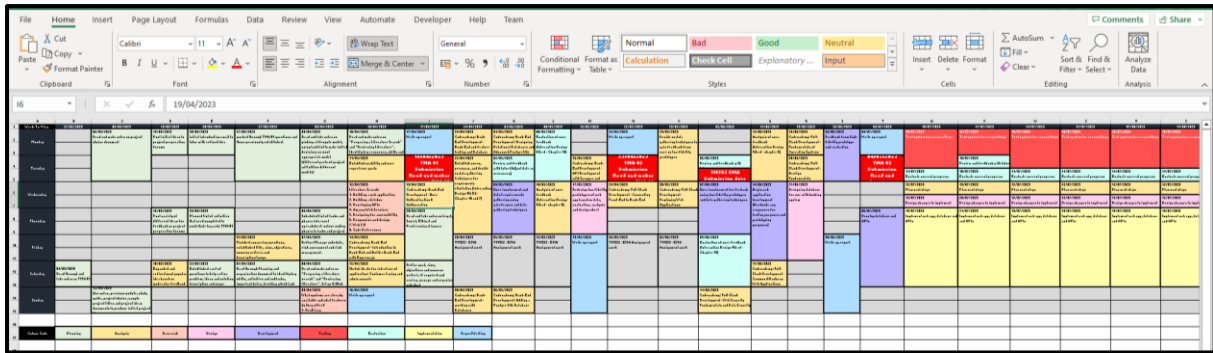


Figure 2: Current iteration of project schedule

## RESOURCES, SKILLS AND METHODS

### Key Resources

- Visual Studio Code: Writing and editing code.
- Microsoft Excel: Scheduling.
- Microsoft Word: Writing notes and reports.
- Git: Version control system for keeping track of progress.
- GitHub: Online repositories for backup and testing.
- StackEdit: Writing markdown documents.
- Interaction Design 4<sup>th</sup> Edition: Research and developing the web application interface.
- OU Library: literature research.
- Users/Stakeholders: Incorporating feedback for design process.
- Codecademy: Learning Database/API development.
- MDN Web Docs/W3School: Reference for HTML, CSS, JavaScript.
- GOV website: GDPR and data research.
- Material Design: Design using pattern languages.

### Key Skills

- Project/Time management
- HTML
- CSS
- JavaScript
- SQL
- Working with local and remote repositories
- Chrome Dev Tools
- Markdown
- Interaction Design
- Web development
- Responsive web design
- Literature search/review
- Report writing
- React
- Redux
- API design/development
- Database design/development



## Risk Assessment and Management

### Chance of Risk:

- Low: It would be considered surprising if it occurred or proved to be the case.
- Medium: It would be disappointing but not surprising if it occurred.
- High: It will very probably prove to be the case.

### Impact of Risk:

- Low: Has negligible effect on the project, a delay of a few days.
- Medium: Has a considerable effect on the project, but it is still likely to succeed.
- High: The project's success is threatened.

### Possibilities for addressing the risk:

- Avoidance – Avoid the risk by taking a course of action that eliminates it.
- Mitigation – insure against the risk and form an appropriate strategy to manage the problem if it does occur.
- Acceptance – Accept the possible negative consequences of the risk.

Project Activity	Risk	Chance	Impact	Method	Plan
<b>Project Planning</b>	Over ambitious, scope too wide reaching, lost quality.	High	Low	Mitigation	Collaborate with tutors and prep forums to iterate ideas.
<b>Project Management</b>	Over estimation of expectations, no schedule, feature creep, poor balance between quality, productivity and time.	Medium	Low	Mitigation	Schedule activities, regularly evaluate progress, manage expectations, define description, scope, aims, objectives, success criteria and essential functionality.
<b>Research and Developing Skills</b>	Wrong material picked that's not relevant, skill not appropriate for project (waster time).	Low	Low	Mitigation	Plan a four stage, thorough literature search, using critical reading techniques such as scanning/skimming, note taking, PROMPT and active reading. Review, evaluation and analysis of materials to choose material with relevance to my project aims and objectives.
<b>Report Writing</b>	Report not written towards and appropriate audience.	Low	Low	Mitigation	Use resources from the OU - "Developing academic English" and other resources to plan and write in the appropriate style for each assignment
<b>Determining Users</b>	Inappropriate users picked.	Low	Medium	Mitigation	Use resources and created personas to establish a range of appropriate users from the perspective of customers and admin management.
<b>Prototyping</b>	Not listening to user's feedback or misunderstanding what the user actually wants.	Low	Medium	Mitigation	Listen to and utilise feedback from user testing and evaluation sessions to identify the most important functionality and incorporate it into the next prototype.
<b>Requirements Gathering</b>	Incorrect and inappropriate techniques used to elicit information.	Low	High	Mitigation	Use resources to identify appropriate techniques specific to my project and apply them in a constructive way. Don't pick a method because it seems convenient.
<b>Development</b>	Poor quality code, inadequate documentation, product doesn't work.	Medium	Low	Accept	Use resources, knowledge gained through developing skills and tutors/forums to the best of ability. If the product doesn't work, evaluate why, what happened, what could have been done better.
<b>Testing</b>	Inadequate management of testing procedures and evaluation of results.	Low	Low	Mitigation	Try to plan a set of comprehensive tests for each iteration before starting work on development to create a well-defined scope and limited range of

					functionality. This will stop feature creep and provide a more focus approach.
<b>Availability or Resources</b>	Might become unavailable during the project, users might have to pull out of providing feedback, paywalls, etc.	Medium	Medium	Accept	Try to incorporate a substantial number of users that is appropriate to the project so you can still continue should a number have to end their involvement. Similarly, try to utilise a range of literature to get different perspectives and cross-reference their arguments so you can continue presenting your ideas with references.
<b>Health</b>	Illness hampering progress.	Medium	Low	Accept	Schedule should incorporate time to review, manage and adjust tasks if illness causes delays.
<b>Technical Limitations</b>	Equipment or knowledge not available causing progress to cause to a crawl.	Low	Medium	Mitigation	Try to find and use a range of resources that provide options to redirect the projects required skills in the event one method isn't feasible.
<b>Quality Control</b>	No testing, evaluation or analysis conducted throughout project.	Low	Medium	Mitigation	Plan appropriate phases into the schedule
<b>Online Work</b>	Weather or other factors disrupting connection.	Low	Medium	Accept	Use local resources to continue work such as version control systems and backup when available. Use mobile data to continue working as much as possible.

# PROJECT WORK

## LITERATURE SEARCH AND REVIEW

### Relational Database Design and Implementation (Harrington, 2016)

An initial first scan indicates this textbook follows a full tutorial from understanding the environment in which databases are used and required for operation, why they are needed and relationships between data, to models, design theory, and implementation. There are a number of case studies providing examples, an introduction to SQL, database security, and beyond.

After a more thorough read, I can see a number of sections that will prove relevant to my project:

- how database requirements are born from a systems analysis and development methodologies (prototyping, spiral, object-oriented analysis and design).
- Effects of poor database design, data modelling independent of specific theoretical data models, entity-relationships and ER diagrams, characteristics of columns and rows, primary keys, data dictionary tables, normalization, performance and partitioning.
- SQL, computer-aided software engineering tools, and case study examples.
- Concurrency control, security, and data quality.

A key element of my project will be the understanding, design and development of a database to store customer details, booking information, staff administration details and access levels. Each of the sections in this material alongside the Codecademy course will provide and extend my knowledge providing a comprehensive, practical and theoretical framework to base my own database on.

---

### Learn PostgreSQL (Ferrari & Pirozzi, 2020)

The interest in this book stems from knowing that part of the Codecademy course which I will be following has sections covering PostgreSQL so this would be an additional resource to compliment learning and cross-reference/confirm knowledge. On first glance, this appears to be aimed at those familiar with databases already and provides an introduction in the first few chapters that explain:

- What PostgreSQL is, its history, versions, dependencies, cluster anatomy, how configuration files are used, command line utilities.
- Management of users and connections, the concept of a “role”, account and group creation.
- Basic and Advance statements.

From chapter 6 and onwards the book delves into seemingly more advanced topics that I will require such as server-side programming, extending the database and physical and logical replication. It also expects prior knowledge of the Linux OS. This resource may be handy later on as a reference and to consolidate knowledge and understanding but may be at a level too high for an introductory reading.

---

### Article - Different Types of Patterns for Online-Booking Systems (Teuber & Forbrig, 2004)

The paper aims to show that by analysing project tasks, users and objects, it's possible to generalise elements that can be applied to other systems to help provide common solutions to reoccurring interface design problems. An online booking system is used as an example. On a first scan some

elements seem questionable (such as how does Paul know early registration causes people quite using the system early? There is no reference), despite this, it might benefit as a starting off point in terms of the types of tasks that might occur for my own system, at least, from a customer's perspective. They abstract user groups from the functionality available - based on the generalised tasks and extrapolate two types of user profiles: First time customers and registered customers. From there they use object-oriented concepts to determine objects, their attributes and relationships between them, finally using these analyses to produce a conceptual design.

Christopher Alexander first proposed the idea of patterns to abstract a recognisable quality and apply it to other designs. However, pattern languages can be more powerful, albeit less common, as they incorporate a network of patterns that reference each other to create a complete structure. (Preece, et al., 2015) A good example of this is [Material Design](#): an adaptable system of guidelines, components and tools to support best practices of user interface design.

---

#### Understanding APIs and RESTful APIs Crash Course (Taulien, 2020)

A very short 40-minute video introduction to understanding what APIs are, how they work and why they're important. While it doesn't provide any guidance on how to build your own API, as a very basic introduction to APIs and request types, with provided examples, it will provide a good starting point and reference for future development of my project.

---

#### RESTful Web API Design with Node.js (Hamadeh, 2016)

A 2-hour video course providing an overview of three RESTful APIs, the steps to build them and their similarities and differences. The three APIs explored are Twitter, Facebook and GitHub. It also looks at best practices to keep APIs secure, maintainable and ensure they perform well. Similar to the previous API Crash course, this feels like more of an introduction to APIs rather than a comprehensive guide or reference on how to build them. With that being said, the Codecademy course uses and implements back-end scripting using Node.js and my project will be developed using HTML, CSS and JavaScript so I feel like this would be a comparatively worthwhile resource to incorporate into my literature.

---

#### Node.js – The Complete Guide (Schwarz Müller, 2019)

A longer more comprehensive video course that guides readers from installing Node.js, working with Express.js, sessions and cookies, authentication, sending emails (may be particularly useful if I am to develop the application's ability to automatically send confirmation and reminder emails), validation, Async requests, REST APIs, WebSockets, deploying applications, testing plus a range of other topics. It provides a substantial amount of practical content alongside the theory and will provide a good reference for my project due to the large amount of content available.

---

#### REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development 2<sup>nd</sup> edition (Doglio, 2018)

Building on the previous three video courses this textbook is a full course beginning with the history of REST, theoretical development, practical API development to the use of Node.js modules to create a RESTful API. It covers requirements gathering to tools section and troubleshooting.

The contents appear to be clearly laid out and structured in a logical way for learning how to develop a REST API starting with an introduction to REST, then following up with API Design best practices, Node.js, Architecting a REST API, working with Node.js modules, REST API planning and development, testing, deploying and finally troubleshooting.

The idea will be to use the video courses and this textbook in conjunction with each other to try and fully understand all the necessary elements required for my web application. While they all contain more advanced materials, they also provide a good reference point for the Codecademy material and alternative points of view and perspectives on how to develop APIs.

---

While this is only an initial literature search and the materials have not been subjected to a thorough read and evaluation I've been able to find value by utilising critical and active reading techniques such as scanning/skimming to help understand the scope and context of each piece of literature and by making notes to justify requiring each of the resources and making sure they are all appropriate and relevant to my project in one form or another. Using PROMPT has also helped to validate the materials and provide me with enough confidence in their legitimacy and value.

### ADDITIONAL WORK

- Setting up a local git repository then connecting it to a GitHub online repository (See Project diary entry #17).
- Background research on available products on the market with a list of common functionalities that can be potentially applied to my own project (See Project Diary Entry #21).
- An initial set of usability and user experience goals (See Project Diary Entry #19).

## REVIEW AND REFLECTION

The initial planning stages have felt very involved and like it has required a lot more study time the suggest 10 hours per week. My project diary can reflect this by acknowledging the fact I have already spent around 60 hours in total working on different aspects of the project. However, I acknowledge this length of time is my own fault as I normally like to take notes by hand and keep a physical copy of work which has substantially decreased my efficiency. It also meant I had to take a significant amount of time copying notes over to a digital format so from this point forward I'll be doing all my work using my digital project diary and backing it up daily to my google drive and GitHub where appropriate. This has also made it more difficult to keep up with my schedule. While I have managed to accomplish everything I set out to, it has made things unnecessarily more stressful.

An important first step has been conducting background research on what is already available, and establishing objectives, success criteria, typical functionality and usability/user experience goals has helped with initial designs from both customer and administrator perspectives. It's also helped define clearer boundaries by understanding what the project isn't - a customer relationship management tool or a sales engagement platform.

My initial approach to tackling this project was to systematically work through all the resources, making notes, relating them to the TMA, and answering the sections in order as I worked through the material. I found this worked well to a point but have since found that sometimes tackling the section as a whole produced a better result such as determining what tasks I need to do, adding them to a schedule, thinking about the risks and how to tackle them, evaluating my work and iterating over the process until I felt I'd made sufficient progress. The project schedule does have specific start and end dates for tasks but now I feel if I don't complete tasks by the identified dates there is still some flexibility to return at a later time.

In terms of developing skills to extend my learning, I have started working through the Codecademy courses and setting up local environments using VS code, Node.js, Git , etc and will incorporate some time to read materials from my literature review more thoroughly. This aspect of the project is the riskiest as I feel like there might be a lot of material to learn, understand and incorporate into my project, however I am lucky enough to have a lot of free time as I don't work so I can effectively treat working on this project as if it were a full-time job. My objective is to spend any free time I have working through, experimenting with and honing these skills to try and ensure I meet my success criteria.

### FEEDBACK/GUIDANCE REQUEST

At the moment I feel confident in the work I have produced so far. I've identified that the most important part of my project is going to be managing the booking system from the perspective of the administrator as they will require a lot more functionality than customers however, I am concerned still that the scope may be rather ambitious based on how much I'll need to extend my knowledge and skill. I'm also apprehensive about trying to identify and access users on the administration side of things and not really sure where to start looking.

## References

- Calendly, 2023. *Calendly Features - Workflows, Integrations, Embeds | Calendly*. [Online]  
Available at: <https://calendly.com/features>  
[Accessed 13 04 2023].
- Doglio, F., 2018. *REST API Development with Node.js : Manage and Understand the Full Capabilities of Successful REST Development*. 2 ed. s.l.:Apress.
- Ferrari, L. & Pirozzi, E., 2020. *Learn PostgreSQL*. s.l.:Packt Publishing.
- G2, 2023. *Highest rated Online Appointment Scheduling Software | G2*. [Online]  
Available at: [https://www.g2.com/categories/online-appointment-scheduling?tab=highest\\_rated](https://www.g2.com/categories/online-appointment-scheduling?tab=highest_rated)  
[Accessed 15 04 2023].
- Google, 2023. *Material Design*. [Online]  
[Accessed 15 04 2023].
- Groove, 2023. *Groove Sales Engagement Platform | Connected Sales Execution*. [Online]  
Available at: <https://www.groove.co/>  
[Accessed 13 04 2023].
- Hamadeh, S., 2016. *REST API Development with Node.js*. s.l.:Packt Publishing.
- Harrington, J. L., 2016. *Relational Database Design and Implementation 4th Edition*. 4th ed. s.l.:Morgan Kaufmann.
- Preece, J., Rogers, Y. & Sharp, H., 2015. In: *Interaction Design: beyond human-computer interaction*. 4 ed. s.l.:Wiley, p. 443.
- RedDiary, 2022. *Training part 1: Introduction - YouTube*. [Online]  
Available at: [https://www.youtube.com/watch?v=mmzC-S\\_vieE&list=PLoKvIZkFOWfDRWCECAiyC8TvCqcH6BdkO&index=2](https://www.youtube.com/watch?v=mmzC-S_vieE&list=PLoKvIZkFOWfDRWCECAiyC8TvCqcH6BdkO&index=2)  
[Accessed 13 04 2023].
- RedDiary, 2023. *ResDiary: Restaurant Booking System*. [Online]  
Available at: <https://resdiary.com/>  
[Accessed 13 04 2023].
- Schwarz Müller, M., 2019. *Node.js - The Complete Guide*, s.l.: Packt Publishing.
- Taulien, K., 2020. *Understanding APIs and RESTful APIS Crash Course*, s.l.: Packt Publishing.
- Teuber, C. & Forbrig, P., 2004. Different Types of Patterns for Online-Booking Systems. *TAMODIA 2004: Proceedings of the 3rd annual conference on Task models and diagrams*, 15 November.pp. 91 - 97.
- YCBM, 2023. *How to set up our scheduling tool online - YouCanBookMe*. [Online]  
Available at: <https://youcanbook.me/howitworks>  
[Accessed 13 04 2023].