

El-Care

Year 4 (Final Year Project)

10-05-2023

**─**

**Amrita Giri, D00226038**

BSc (Hons) Computing in Cloud Computing, 2023

**Dundalk Institute of Technology**

Copyright Notice

**Supervisor :** Tony Mc Carron

**Second Reader :** Dr Peadar Grant

**ACKNOWLEDGEMENTS**

I am grateful and full of gratitude to my supervisor Professor Tony Mc Carron and my second reader Dr. Peadar Grant, for their unwavering support, guidance, and invaluable insights throughout the process of writing this thesis and yearlong project.

Finally, I am forever indebted to my family , my classmates’ and close friends for their unwavering love, support, and understanding throughout my academic pursuits.

Best Wishes

**DECLARATION**

I hereby declare that the work described in this project is, except where otherwise stated, entirely my own work and has not been submitted as part of any degree at this or any other Institute/University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Amrita Giri

2023

Table of Contents

[ABSTRACT 8](#_Toc134741276)

[LITERATURE REVIEW 9](#_Toc134741277)

[Project Aim & Objective 9](#_Toc134741278)

[Market Research 11](#_Toc134741279)

[Market Analysis 11](#_Toc134741280)

[Competitor Analysis 11](#_Toc134741281)

[User Needs Assessment 12](#_Toc134741282)

[Industry Trends 12](#_Toc134741283)

[Regulatory Environment 13](#_Toc134741284)

[SWOT Analysis 13](#_Toc134741285)

[Market Entry Strategy 14](#_Toc134741286)

[Why Hosting the Application by Cloud Computing for Any Project Beneficial ? 15](#_Toc134741287)

[Cost Savings 15](#_Toc134741288)

[Security 15](#_Toc134741289)

[Flexibility 16](#_Toc134741290)

[Mobility 16](#_Toc134741291)

[Insight 16](#_Toc134741292)

[Rise in Collaboration 16](#_Toc134741293)

[Quality Management 17](#_Toc134741294)

[Disaster Prevention 17](#_Toc134741295)

[Loss Recovery 17](#_Toc134741296)

[Automatic Software Updating and Maintenance 17](#_Toc134741297)

[Competitive Edge 17](#_Toc134741298)

[Sustainability 18](#_Toc134741299)

[Scalability 18](#_Toc134741300)

[Reliability 18](#_Toc134741301)

[Global Reach 18](#_Toc134741302)

[Integration and Collaboration 18](#_Toc134741303)

[Models of Cloud Services 18](#_Toc134741304)

[IAAS (Infrastructure-as-a-Service) 19](#_Toc134741305)

[PAAS (Platform-as-a-Service) 19](#_Toc134741306)

[SAAS (Software-as-a-Service) 20](#_Toc134741307)

[FAAS(Function-as-a-Service) 21](#_Toc134741308)

[Cloud Use Cases 21](#_Toc134741309)

[ Edge computing and IoT 21](#_Toc134741310)

[ AI and machine learning 22](#_Toc134741311)

[ Serverless and event-driven architectures 22](#_Toc134741312)

[ Quantum computing 22](#_Toc134741313)

[ 5G and cloud-native networking 22](#_Toc134741314)

[ Decentralized and distributed cloud 22](#_Toc134741315)

[ Enhanced security and privacy 22](#_Toc134741316)

[ Green and sustainable cloud computing 23](#_Toc134741317)

[ Industry-specific cloud solutions 23](#_Toc134741318)

[Amazon Web Services 23](#_Toc134741319)

[Google Cloud Platforms 24](#_Toc134741320)

[Microsoft Azure 25](#_Toc134741321)

[Azure vs AWS vs GCP 25](#_Toc134741322)

[Market share 28](#_Toc134741323)

[Services 29](#_Toc134741324)

[Pricing 29](#_Toc134741325)

[Compliance and security 30](#_Toc134741326)

[Database Technology 30](#_Toc134741327)

[Relational databases 31](#_Toc134741328)

[MySQL 31](#_Toc134741329)

[NoSQL databases 32](#_Toc134741330)

[Firebase (Realtime Database and Fire store) 32](#_Toc134741331)

[MongoDB 33](#_Toc134741332)

[Frontend Technology 33](#_Toc134741333)

[Backend Technologies 34](#_Toc134741334)

[ Programming languages 35](#_Toc134741335)

[ Web frameworks 35](#_Toc134741336)

[ Caching systems 35](#_Toc134741337)

[ Message brokers 35](#_Toc134741338)

[ Server and deployment technologies 35](#_Toc134741339)

[Visual Studio Code 36](#_Toc134741340)

[PROPOSED SOLUTION 38](#_Toc134741341)

[Design 38](#_Toc134741342)

[Basic Functionality Brief: 38](#_Toc134741343)

[Design Snippets and Images 38](#_Toc134741344)

[Implementation 40](#_Toc134741345)

[Database (MySQL) 40](#_Toc134741346)

[Frontend Technologies (Bootstrap and HTML CSS) 41](#_Toc134741347)

[Backend Technologies 46](#_Toc134741348)

[Cloud Service and Connections 49](#_Toc134741349)

[Testing /Evaluation 54](#_Toc134741350)

[Demonstration Record 55](#_Toc134741351)

[CONCLUSIONS 63](#_Toc134741352)

[original aims and objectives of the project 63](#_Toc134741353)

[Experience of undertaking an individual project 65](#_Toc134741354)

[Evidence that student understands and is aware of the limitations of the work conducted and would be able to develop it further in future work 65](#_Toc134741355)

[APPENDICES 66](#_Toc134741356)

[REFERENCES 67](#_Toc134741357)

**Table of Figures**

[Figure 1 : Volunteering at Social Services 10](#_Toc134741867)

[Figure 2 : El-Care (For your Loved Ones) 11](#_Toc134741868)

[Figure 3 : Market Analysis 12](#_Toc134741869)

[Figure 4 : Competitor Analysis 13](#_Toc134741870)

[Figure 5 : User Requirement 13](#_Toc134741871)

[Figure 6 : Industry Trends 14](#_Toc134741872)

[Figure 7 : Law and Regulation 14](#_Toc134741873)

[Figure 8 : SWOT Analysis 15](#_Toc134741874)

[Figure 9 : Market Entry Strategy 15](#_Toc134741875)

[Figure 10 : Advantages of Cloud 16](#_Toc134741876)

[Figure 11 : Models of Cloud Services 20](#_Toc134741877)

[Figure 12 : IAAS 20](#_Toc134741878)

[Figure 13 : PAAS 21](#_Toc134741879)

[Figure 14 :SAAS 21](#_Toc134741880)

[Figure 15 : FAAS 22](#_Toc134741881)

[Figure 16 : AWS 24](#_Toc134741882)

[Figure 17 : GCP 25](#_Toc134741883)

[Figure 18 : Azure 26](#_Toc134741884)

[Figure 19 : AWS vs GCP vs Azure 28](#_Toc134741885)

[Figure 20 : AWS vs GCP vs Azure Comparison 29](#_Toc134741886)

[Figure 21 : AWS vs GCP vs Azure Market Share 29](#_Toc134741887)

[Figure 22 : AWS vs GCP vs Azure Services 30](#_Toc134741888)

[Figure 23 : AWS vs GCP vs Azure Pricing 30](#_Toc134741889)

[Figure 24 : AWS vs GCP vs Azure Compliance and Security 31](#_Toc134741890)

[Figure 25 : Relational Database 32](#_Toc134741891)

[Figure 26 : MySQL 32](#_Toc134741892)

[Figure 27 : NoSQL DB 33](#_Toc134741893)

[Figure 28 : Firebase 33](#_Toc134741894)

[Figure 29 : MongoDB 34](#_Toc134741895)

[Figure 30 : Frontend Technologies 34](#_Toc134741896)

[Figure 31 : Backend Technologies 35](#_Toc134741897)

[Figure 32 : Visual Studio Code 37](#_Toc134741898)

[Figure 33 : Basic Architecture Diagram 39](#_Toc134741899)

[Figure 34 : Technology based Architecture 40](#_Toc134741900)

[Figure 35 : Technologies Used 40](#_Toc134741901)

[Figure 36 : UML Class Diagram 40](#_Toc134741902)

[Figure 37 : PhpMyAdmin 41](#_Toc134741903)

[Figure 38 : Database Tables 41](#_Toc134741904)

[Figure 39 : Bootstrap 42](#_Toc134741905)

[Figure 40 : Bootstrap Layout 43](#_Toc134741906)

[Figure 41 : AWS Management Console Page 52](#_Toc134741907)

[Figure 42 : Instance Created for the Website 53](#_Toc134741908)

[Figure 43 : Putty Login with Instance Details 53](#_Toc134741909)

[Figure 44 : Putty Login with Private Key 54](#_Toc134741910)

[Figure 45 : .pem and .ppk files for the Instance key 54](#_Toc134741911)

[Figure 46 : Login as Ubuntu (Linux Instance) 54](#_Toc134741912)

[Figure 47 : Website Running on Instance 55](#_Toc134741913)

[Figure 48 : DB running on Instance 55](#_Toc134741914)

[Figure 49 : Website Home Page 57](#_Toc134741915)

[Figure 50 : About Us Page 57](#_Toc134741916)

[Figure 51 : About Us Page Continued 57](#_Toc134741917)

[Figure 52 : Testimonials Page 58](#_Toc134741918)

[Figure 53 : Our Team Page 58](#_Toc134741919)

[Figure 54 : Contact Us Page 59](#_Toc134741920)

[Figure 55 : Services Page 59](#_Toc134741921)

[Figure 56 : Services Page Continued 59](#_Toc134741922)

[Figure 57 : Booking Form Page 60](#_Toc134741923)

[Figure 58 : Booking Confirmation Page 60](#_Toc134741924)

[Figure 59 : Admin Login Page 61](#_Toc134741925)

[Figure 60 : User Registration Page 61](#_Toc134741926)

[Figure 61 : User Login Page 62](#_Toc134741927)

[Figure 62 : Employee Login Page 62](#_Toc134741928)

[Figure 63 : Employee Login Display List 63](#_Toc134741929)

[Figure 64 : Admin Login Display List 63](#_Toc134741930)

# 

# ABSTRACT

I have been volunteering in old age homes and disability schools since a very young age and noticed a lack of proper technological systems for these two sectors. I believe that these sectors should also benefit from advancements in technology and the digital era. We have identified common problems faced by individuals who are unable to take care of their parents while away in another city or part of the world. To solve this problem, I propose a project called El-Care, which would act as a proxy for individuals to have everything their loved ones need in their hands.



Figure 1 : Volunteering at Social Services

The El-Care project consists of two distinct parts: the front-end and the backend. The front-end is a web application that communicates with clients and develops information from the backend into a comprehensible configuration. The backend is a cloud-supported project that can read and write any type of data to the website. This is justified using various technologies such as JavaScript, Visual Studio, AWS Cloud, PHP, GitHub, HTML/CSS, and Bootstrap for the development of this project.

# LITERATURE REVIEW

## Project Aim & Objective

The primary aim is for a user-friendly, accessible, and efficient online service booking website specifically tailored for elderly individuals. This website will provide a comprehensive range of services and by centralizing these services in one platform, the project seeks to streamline the booking process, enhance communication between service providers and elderly users, and foster a supportive community for seniors to easily access and manage essential services, ultimately contributing to their well-being and social inclusion.



Figure 2 : El-Care (For your Loved Ones)

1. To conduct thorough research and identify the essential services required by the elderly population, taking into consideration their unique needs and preferences.
2. To design a user-friendly and intuitive interface for the elderly service booking platform, prioritizing accessibility, and ease of navigation for senior users with varying levels of digital literacy.
3. To develop a secure and reliable platform that ensures the privacy and safety of users' personal information and provides an efficient booking process for various services.
4. To implement user feedback mechanisms with chat systems that allow continuous improvement of the platform's features and functionality, based on users' experiences and suggestions.

## Market Research

### Market Analysis



Figure 3 : Market Analysis

1. **Target Market:** mostly elderly individuals aged 65 and above, as well as their family members or caregivers who may be responsible for managing their daily needs.
2. **Market Size and Segmentation:** analyze population data, aging trends, geographic location, income levels, living arrangements, specific service requirements, and projected growth of the elderly population in the region or country.

### Competitor Analysis



Figure 4 : Competitor Analysis

1. **Identifying Major Competitors :** in the elderly service booking industry, both online and offline. Assess their market share, strengths, weaknesses, and unique selling propositions.
2. **Investigating their Business :** their pricing strategies, range of services, target audience, and user experience to identify gaps or opportunities for differentiation.

### User Needs Assessment

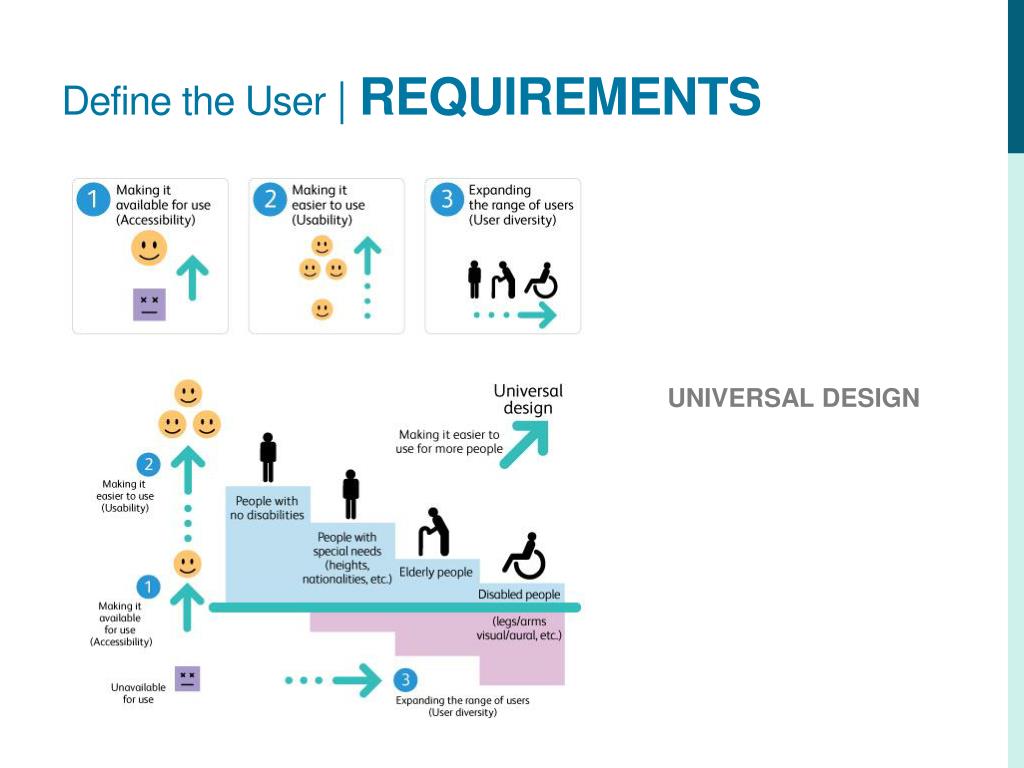


Figure 5 : User Requirement

1. **User Information Gathering :**Conducting surveys, interviews, or focus groups with elderly individuals and their caregivers to understand that they have preferred needs , preferences, and challenges of accessing and booking services.
2. **Gather Insights :** into their level of digital literacy, accessibility requirements, and the factors that influence their choice of service providers.

### Industry Trends



Figure 6 : Industry Trends

1. **Analyzing Trends:** current and emerging trends in the elderly care market, such as the increasing demand for home care services, the integration of technology in care provision, and the growing emphasis on personalized care.
2. **Assessing Impact:** the potential impact of these trends on the elderly service booking industry and identify opportunities for innovation or adaptation.

### Regulatory Environment



Figure 7 : Law and Regulation

1. **Research Regulations:** relevant regulations and guidelines governing the elderly care industry, such as licensing requirements, data privacy, and safety standards.
2. **Ensuring Law Amendment:** that the proposed platform complies with all applicable laws and standards, and factors in any potential regulatory changes when developing the website.

### SWOT Analysis

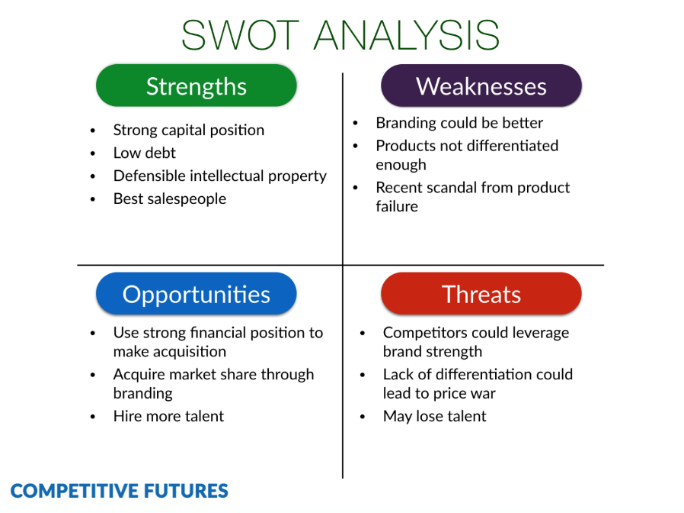


Figure 8 : SWOT Analysis

1. **SWOT:** Identify the strengths, weaknesses, opportunities, and threats related to the proposed elderly service booking website.
2. **Data Analysis :** Use this analysis to inform the platform's development strategy, marketing approach, and risk mitigation plans.

### Market Entry Strategy

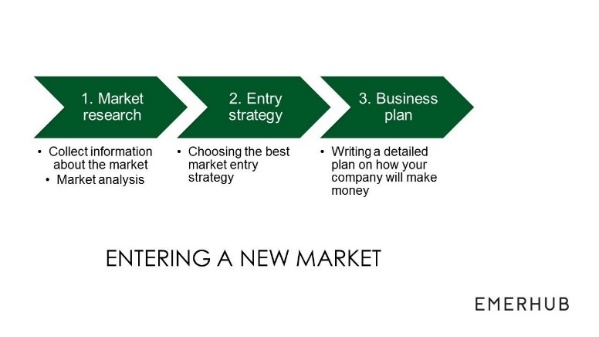


Figure 9 : Market Entry Strategy

1. **Developing Market Strategy:** a market entry strategy, including the selection of an initial target market segment, pricing strategy, promotional tactics, and distribution channels.
2. **Potential Partnership :** Consider potential partnerships with relevant organizations, such as senior centers, healthcare providers, or community organizations, to facilitate a successful market entry and foster long-term growth.

## Why Hosting the Application by Cloud Computing for Any Project Beneficial ?

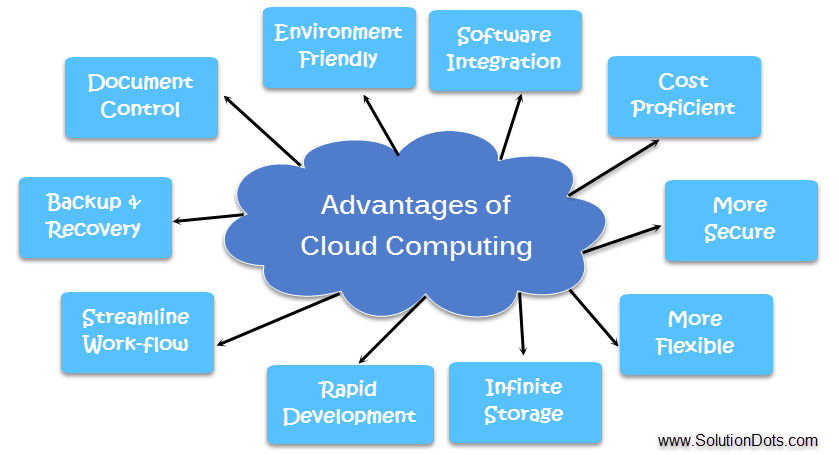


Figure 10 : Advantages of Cloud

Cloud is the amalgamation of virtualization resources at a cloud data center that is run at big infrastructure level by leading cloud service providers in the world. The resources can vary ranging from servers (including physical, virtual and hardware servers) , data collection and storage , operating system software and networking and security. These are the services that can be used for a monthly subscription fee or just charges for services used.

Whether it is work from home in a laptop or a mobile a recent survey showed that 92% of the organizations today have been seen using all the services provided by cloud server providers. So, do we at home unknowingly with google Gmail, google drive , Netflix or drop box. [22]

### Cost Savings

Paying for exactly the space and resources used and needed showed result in lower costs and higher returns.

### Security

Statistics of Rapid Scale said that there was a 94% of businesses improvement and meeting requirements in security after organizations changed to the cloud. Encryption of data used amped up in cloud made it less accessible for hackers to view data. [27]

### Flexibility

Extra bandwidth is easier in cloud rather than expensive update in local server IT Infrastructure proving significant difference to the overall efficiency of many organizations. InformationWeek mentioned that 65% said they were more able to meet the business demands of their clients. [28]

### Mobility

Cloud computing allows our billion mobile users to stay in the loop of accessing corporate data via their smartphones making work from home a very safe work environment

### Insight

Our ability to critically see our cloud analytics of our data with an eagle’s eye hence making tracking and customized report analysis easier.

E.g.: a beverage company increased profits by about $2 million a year but also were able to cut down $195,000 in their staff cost just by introducing cloud technologies in their company .[28]

### 

### Rise in Collaboration

Working in a team is made easier with cloud services because of teamwork social gatherings to connect employees across your organization, therefore increasing their pique and work engagement.

### Quality Management

In an organization where the large teams and team worker must access all kinds of data in between projects and project works this kind of data handling can always result in human error and this can be solved by having a clear set of data in a document with a consistent template in the cloud. This also helps in maintaining a template that is of same consistency on the cloud to avoid any mixing of work or confusion or dilution of data

### Disaster Prevention

This industry is extremely fragile in cases where having even a downtime can have some bad effects on productivity, revenue, and brand reputation. Hence cloud managed to provide a massive disaster recovery to the extent that many companies got saved from running into massive losses or business failures with clients.

### Loss Recovery

The risk of losing data is huge in today’s day and age especially the ones that are saved locally but the risk is reduced with cloud for all the information that can be uploaded online remains safe and can be accessed by any computer or laptop that has internet connection

### Automatic Software Updating and Maintenance

The automatic updating and refreshing of software and avoiding spending time and money on external experts. It was also seen that half of the world population used less external hardware and software appliances after using Cloud resources as a benefit over them

### Competitive Edge

It is always an edge over the competitors when you manage to learn a technology that is relatively new cause you have an edge learning with experience by the time they catch up. The advantage of this approach showed that 77% had competitive advantage and significance. [28]

### Sustainability

Hosting on cloud resulted in carbon foot printing being reduced and there was a wastefulness at almost every level of a business proving to be more environment friendly

### Scalability

Cloud Computing has a great quality of being able to provide resources that can be demanded for a scale up or down depending on company needs

### Reliability

Cloud service providers offer robust infrastructures with multiple layers of redundancy, ensuring high availability and minimizing downtime.

### Global Reach

Cloud hosting allows the website to be easily accessible from anywhere in the world, ensuring that elderly users and their caregivers can access the platform and book services regardless of their location.

### Integration and Collaboration

Cloud computing makes it easier to integrate the website with other software, tools, and applications, such as customer relationship management (CRM) systems, billing software, and data analytics tools. This can help streamline operations, improve communication with service providers, and enhance the overall user experience.

## Models of Cloud Services

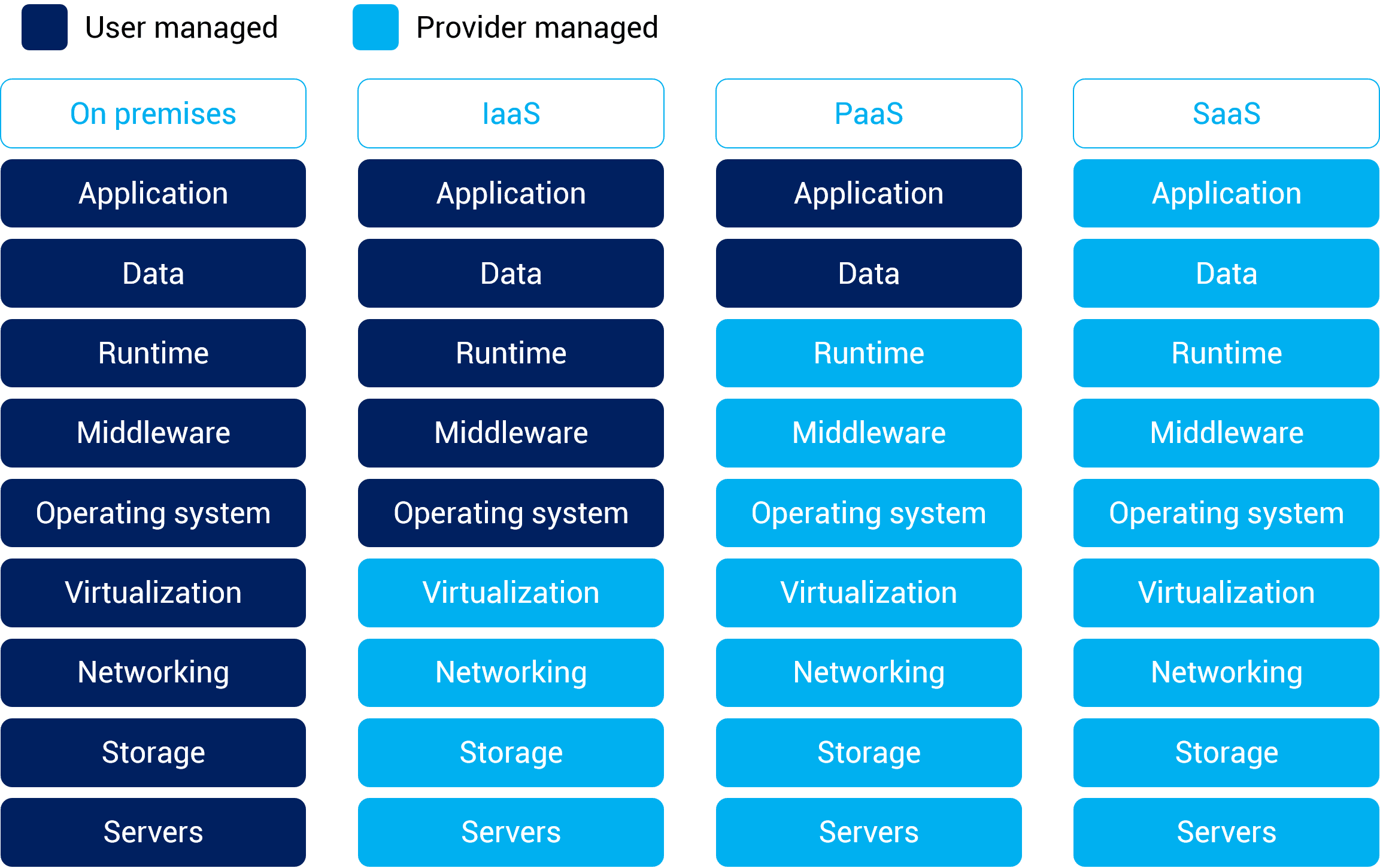


Figure 11 : Models of Cloud Services

### IAAS (Infrastructure-as-a-Service)

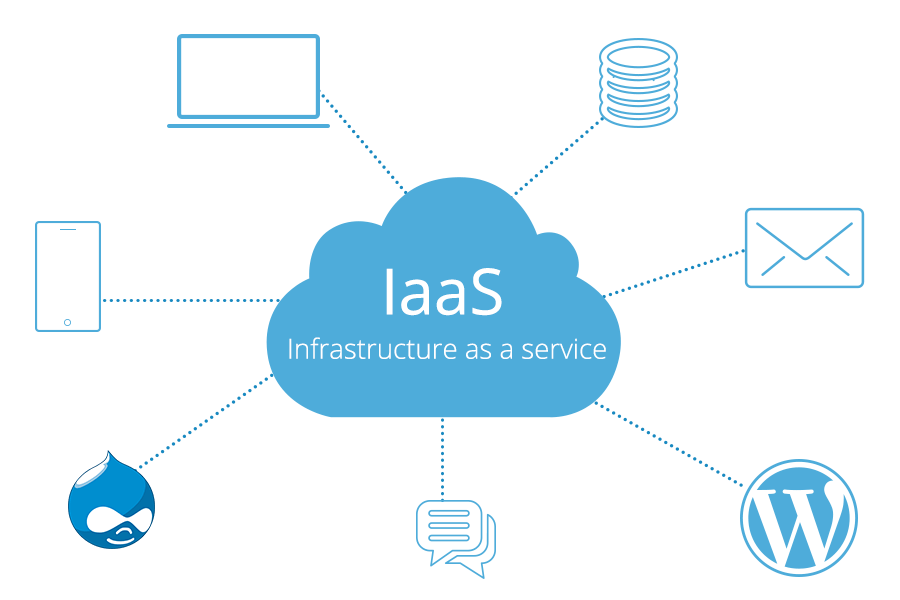


Figure 12 : IAAS

1. virtualized computing resources over the internet.
2. virtual machines (VMs), storage, networking, and other necessary computing resources on pay basis
3. AWS, GCP and Azure
4. services and tools can be customized to suit the specific needs of a business or organization.

### PAAS (Platform-as-a-Service)

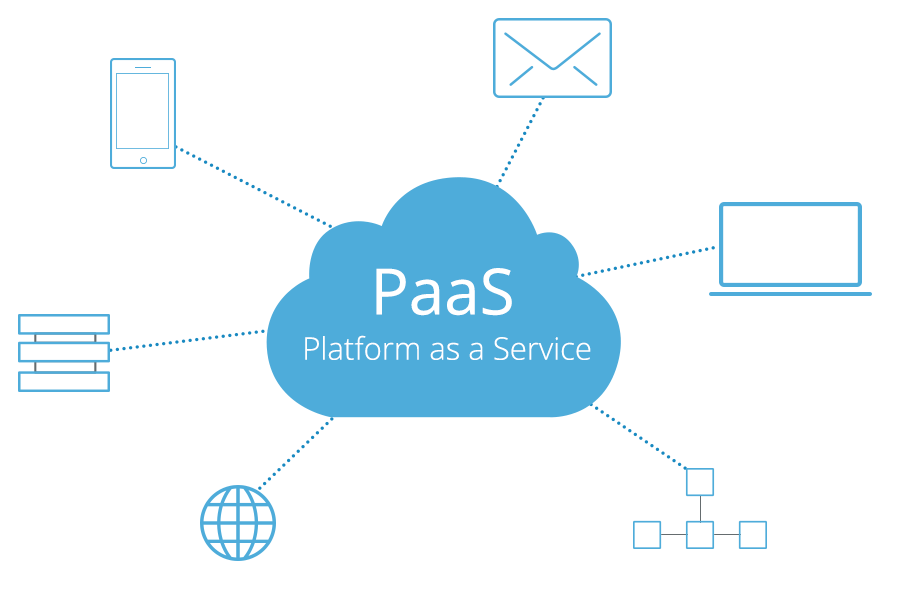


Figure 13 : PAAS

1. developers to build, deploy, and manage applications without infrastructure complexities.
2. PaaS offerings typically include tools and services for developing, testing, deploying, and maintaining applications.
3. Heroku, Google App Engine, Microsoft Azure App Service, and IBM Cloud Foundry.
4. cater to different programming languages and frameworks

### SAAS (Software-as-a-Service)



Figure 14 :SAAS

1. eliminates the need to install and maintain the software on their local devices.
2. host, maintain, and manage the software through a web browser, usually via a subscription model.
3. SaaS offers several benefits like Lower upfront costs, Easy access, Scalability, Automatic updates, Simplified integration
4. Some well-known SaaS examples include Salesforce (a customer relationship management platform), Microsoft Office 365 (productivity suite), Google Workspace (collaboration and productivity tools), and Slack (team communication and collaboration).

### FAAS(Function-as-a-Service)

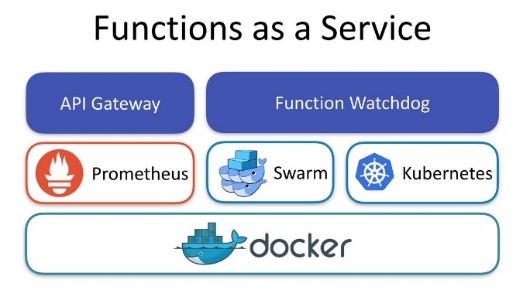


Figure 15 : FAAS

1. build and run applications without servers and infrastructure
2. section of code performs specific tasks, while the cloud provider takes care of the operational aspects like scaling, patching, and capacity management.
3. AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, and IBM Cloud Functions.

## Cloud Use Cases

As cloud computing continues to evolve, new use cases and applications will emerge. Here are some potential future cloud use cases:

### 

### Edge computing and IoT

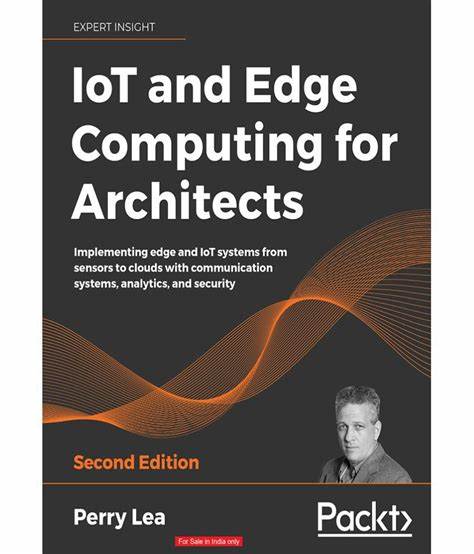


Figure 16 : Edge computing and IoT

Edge computing will become more important to process data closer to the source as IoT devices grow, allowing for real-time data processing and analytics.

### AI and machine learning



Figure 17 : AI and machine learning

Cloud providers will expand their offerings for AI and machine learning, providing access to powerful algorithms and tools to enable businesses to leverage AI.

### Serverless and event-driven architectures

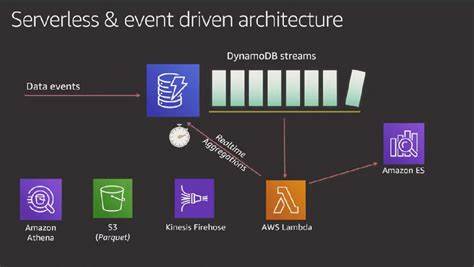


Figure 18 : Serverless and event-driven architectures

Serverless computing and event-driven architectures will grow, enabling organizations to build efficient and scalable applications with minimal infrastructure management overhead.

### Quantum computing

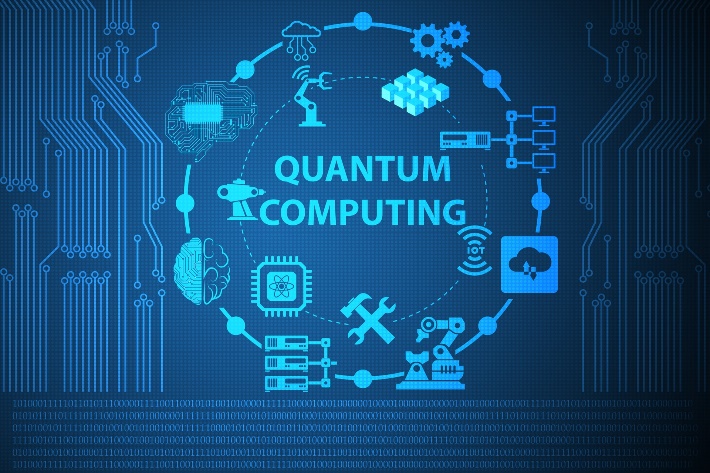


Figure 19 : Quantum computing

Quantum computing may mature to allow cloud providers to offer quantum computing resources to customers, enabling them to solve complex problems currently intractable for classical computers.

### 5G and cloud-native networking



Figure 20 : 5G and cloud-native networking

The rollout of 5G networks will facilitate faster data transfer and improved connectivity, enabling new cloud use cases.

### Decentralized and distributed cloud

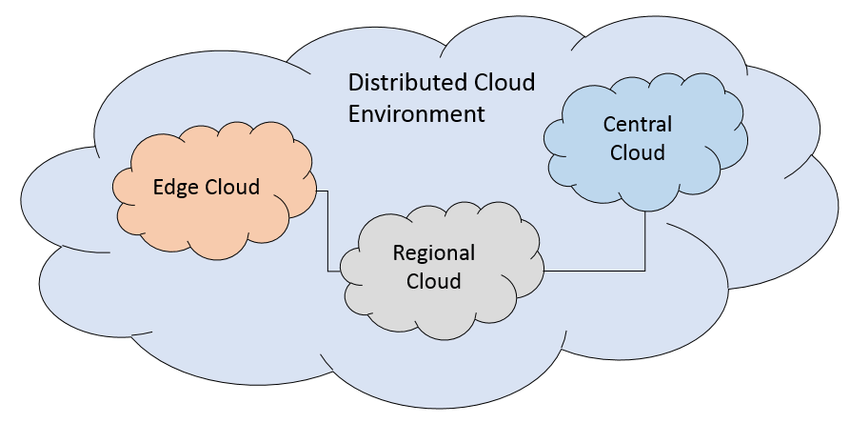


Figure 21 : distributed cloud

Decentralized and distributed cloud solutions may gain traction as alternatives to centralized cloud providers, offering increased resilience, privacy, and control over data.

### Enhanced security and privacy



Figure 22 : Enhanced security and privacy

Cloud providers will develop new tools and services to help organizations protect their data and comply with regulations.

### Green and sustainable cloud computing



Figure 23 : Green and sustainable cloud computing

Cloud providers will focus on sustainable and energy-efficient solutions, such as utilizing renewable energy sources and developing energy-efficient data centers.

### Industry-specific cloud solutions

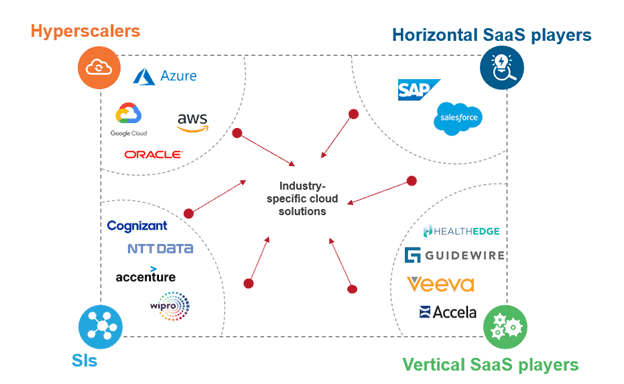


Figure 24 : Industry-specific cloud solutions

Cloud providers will develop more industry-specific solutions tailored to the unique needs and requirements of sectors such as healthcare, finance, manufacturing, and education.

Cloud computing will continue to play a critical role in enabling organizations to innovate, scale, and adapt to changing business environments as technology evolves.

## Amazon Web Services



Figure 25 : AWS

1. a broad range of services and tools for organizations to build, deploy, and manage applications.
2. It has a global infrastructure of data centers and availability zones across regions worldwide, enabling organizations to build highly available and scalable applications while minimizing latency for end-users.
3. AWS offers services such as computing, storage, databases, networking, security, identity, analytics, machine learning and AI, and developer tools, catering to various industries.

## Google Cloud Platforms



Figure 26 : GCP

1. services that offer infrastructure, platform, software solutions for organizations to build, deploy, and manage applications.
2. GCP offers key services and features such as Google Compute Engine, Google Kubernetes Engine, Cloud Functions, Google Cloud Storage, Persistent Disk, Cloud SQL, Cloud Fire store, and Cloud Load Balancing.
3. GCP is built on the same infrastructure that powers Google's own services, making it a reliable and powerful cloud computing platform.

## Microsoft Azure

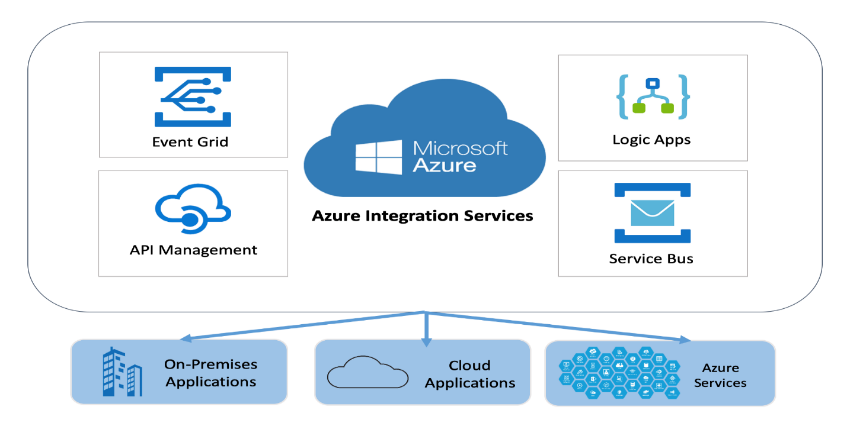


Figure 27 : Azure

1. services and tools for building, deploying, and managing applications.
2. Offers IaaS, PaaS, SaaS to cater to different customer needs.
3. It has a global infrastructure with data centers and availability zones in multiple regions, enabling organizations to build highly available and scalable applications.
4. key services computing, storage, database, networking, security and identity, analytics, machine learning and AI, and developer tools.

## Azure vs AWS vs GCP

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **GCP** | **AWS** | **Azure** |
| **Containerization** | Kubernetes (orchestration) | Relatively new | Offers platforms but not as good as GCP |
| **Market Position** | 3rd on market share | On top due to services provided | 2nd in market share |
| **Support Material** | Mastery possible with documentation/ videos | Good practical training needed to master AWS | No proper documentation hence self-study needed to master Azure |
| **Offerings** | 2nd option than AWS as both offer similar platforms | Services and support given to large organization | Companies only prefer azure for windows or can do it without outside help |
| **Global Marketing** | Global outreach is comparatively less as they don’t market as much | Due to marketing efforts, it known as the best cloud platform by large and small organizations | Azure is known but used less because AWS is a competitor. AWS policies are easier to follow than Azures |
| **Integration** | Gmail, YouTube and all other google services for seamless experience with cloud services | Users are happy with services also they are rendered | Microsoft integrated tools and software |
| **Open Sourced** | Not open sourced but some services are free. Provides portability of its services | Not open sourced need to pay as you use described accordingly | Open Sourced so users use it |
| **Tools Management** | Managed well but less services | Effective tool management and very vast services offered | Not proper management of tools |
| **Designing** | native environment design, offer/discounts/flexible contracts offered | For large storage and network usage | Big data and ML |
| **Focus Area** | DevOps with Docker and Kubernetes | different services on different platforms | Many features set mainly focusing on hybrid cloud |

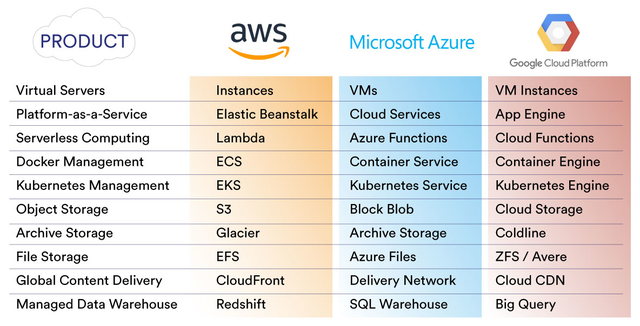


Figure 28 : AWS vs GCP vs Azure

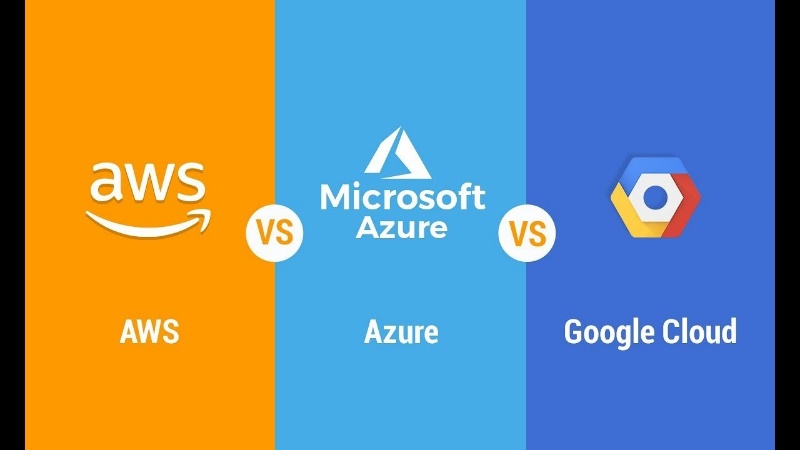


Figure 29 : AWS vs GCP vs Azure Comparison

AWS, GCP, and Azure are three leading cloud computing platforms, each with its own strengths and offerings. Here is a comparison of the three in terms of market share, services, and other factors:

### Market share

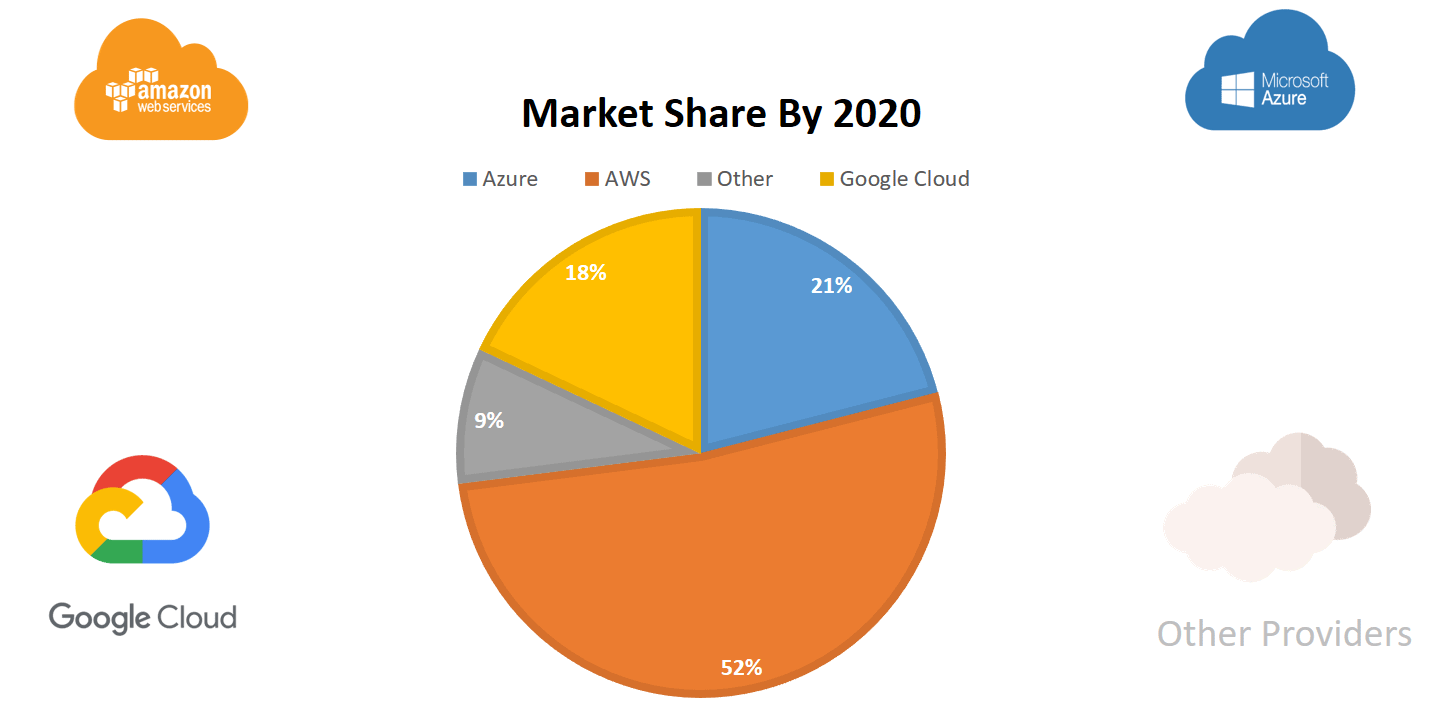


Figure 30 : AWS vs GCP vs Azure Market Share

1. AWS is the market leader with the largest share and has been in the cloud computing space the longest, providing them with a more mature and extensive set of services.
2. Azure is the second-largest player, benefiting from Microsoft's existing enterprise customer base and integration with other Microsoft products.
3. GCP is the third-largest platform, known for its innovation and focus on developer-friendly tools, as well as its machine learning and data analytics capabilities.

### Services

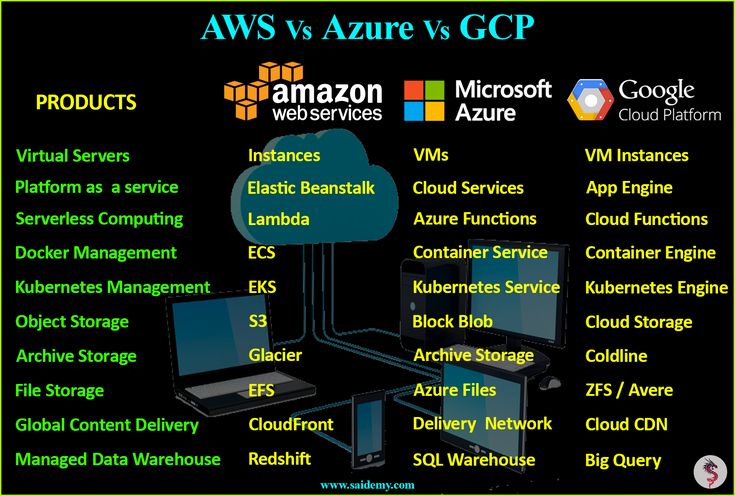


Figure 31 : AWS vs GCP vs Azure Services

1. AWS has the broadest range of services and features, which can be beneficial for organizations looking for the most extensive set of options.
2. Azure's tight integration with other Microsoft products and services can be advantageous for organizations already using Microsoft solutions.
3. GCP is particularly strong in data analytics and machine learning, leveraging Google's expertise in these areas.

### Pricing

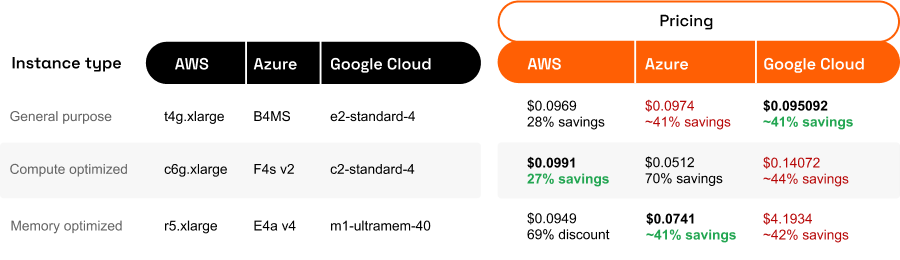


Figure 32 : AWS vs GCP vs Azure Pricing

1. All three platforms use pay per use only for the resources they consume.
2. AWS and Azure offer similar pricing structures, while GCP tends to be slightly more cost-effective for some services, particularly in data storage and transfer.
3. All three platforms provide various cost optimization tools and options, such as reserved instances, committed use contracts, or savings plans.

### Compliance and security

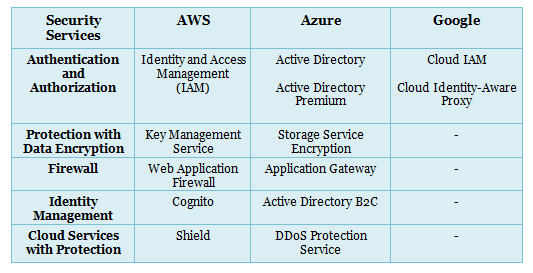


Figure 33 : AWS vs GCP vs Azure Compliance and Security

1. AWS, GCP, and Azure all prioritize security and compliance, offering all services to help organizations secure their data and applications.
2. All three platforms comply with numerous industry standards and certifications, but specific compliance offerings may vary by platform.

Ultimately, the choice between AWS, GCP, and Azure depends on an organization's specific requirements, existing infrastructure, and familiarity with the platforms. Many organizations opt for a multi-cloud strategy, using more than one cloud provider to take advantage of each platform's unique strengths and features.

## Database Technology

Databases are organized collections of data, typically stored and accessed electronically from a computer system. They are essential for managing, storing, and retrieving large amounts of structured and semi-structured data efficiently. Databases enable quick access to relevant information, facilitating data manipulation and analysis.

### Relational databases

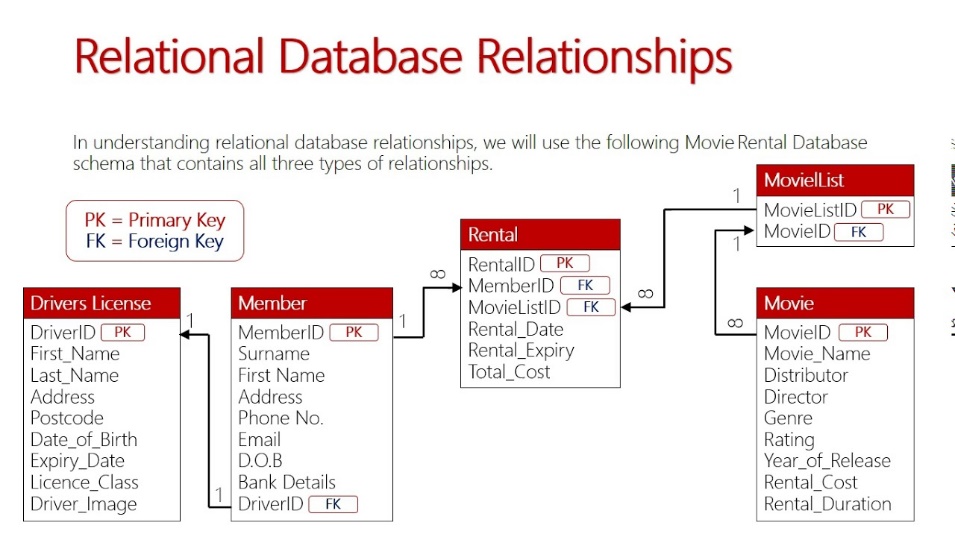


Figure 34 : Relational Database

1. They store data in tables with rows and columns, and relationships between tables are defined through primary and foreign keys.
2. Examples include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.

### MySQL



Figure : MySQL

MySQL is an open-source relational database management system (RDBMS) that has been widely adopted for its performance, reliability, and ease of use. It uses SQL as the query language and is suitable for a wide range of applications. Some reasons for its popularity include:

1. Mature and well-established
2. Easy to set up and use
3. Cost-effective
4. Compatibility
5. Scalability and performance

### NoSQL databases

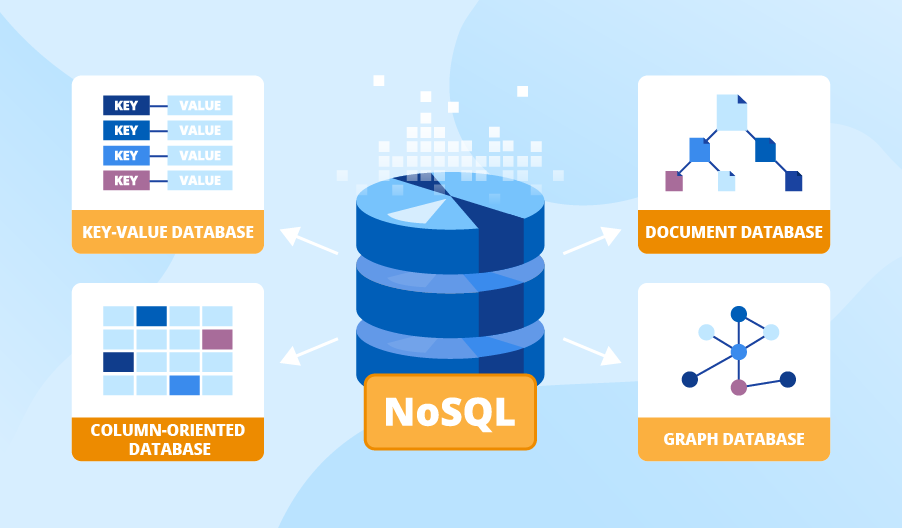


Figure 36 : NoSQL DB

1. These databases are designed for handling unstructured or semi-structured data and do not rely on a fixed schema.
2. Examples include MongoDB (document-based), Cassandra (column-family), Redis (key-value), and Neo4j (graph).

### Firebase (Realtime Database and Fire store)



Figure 37 : Firebase

Firebase is a cloud-based platform developed by Google that offers a suite of backend services, including Realtime Database and Fire store, which are NoSQL databases.

1. Realtime Database
2. Fire store

### MongoDB



Figure 38 : MongoDB

MongoDB is a popular open-source NoSQL database that stores data in a flexible, JSON-like format called BSON. It is designed to handle unstructured data and can scale horizontally across multiple nodes. Some reasons for MongoDB's popularity include:

1. Schema-less design
2. Scalability
3. High performance
4. Rich query language
5. Robust ecosystem

## Frontend Technology



Figure 39 : Frontend Technologies

Front-end technologies are the tools and frameworks used to create the user interface (UI) and user experience (UX) of a website or web application. They are responsible for the layout, design, and interactivity of a site.

1. HTML (Hyper Text Markup Language)
2. CSS (Cascading Style Sheets)
3. JavaScript
4. Responsive Design
5. CSS Frameworks
6. JavaScript Libraries & Frameworks
7. CSS Preprocessors
8. Build Tools and Task Runners
9. Version Control Systems
10. Testing and Debugging Tools
11. Performance Optimization

## Backend Technologies

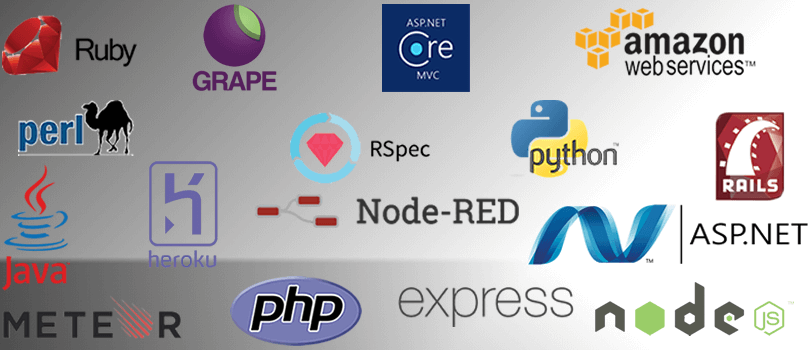


Figure 40 : Backend Technologies

Backend technologies refer to the server-side technologies that power web applications, mobile apps, and other software systems. They are responsible for processing, storing, and managing data, as well as ensuring smooth communication between the client-side (frontend) and the server-side (backend).

### Programming languages

1. Python
2. JavaScript
3. Java
4. PHP
5. Ruby
6. C#

### Web frameworks

1. Django (Python)
2. Flask (Python)
3. Express.js (JavaScript/Node.js)
4. Spring Boot (Java)
5. Laravel (PHP)
6. Ruby on Rails (Ruby)

### Caching systems

1. Redis
2. Memcached

### Message brokers

1. RabbitMQ
2. Apache Kafka

### Server and deployment technologies

1. Nginx
2. Apache HTTP Server
3. Docker
4. Kubernetes

## Visual Studio Code

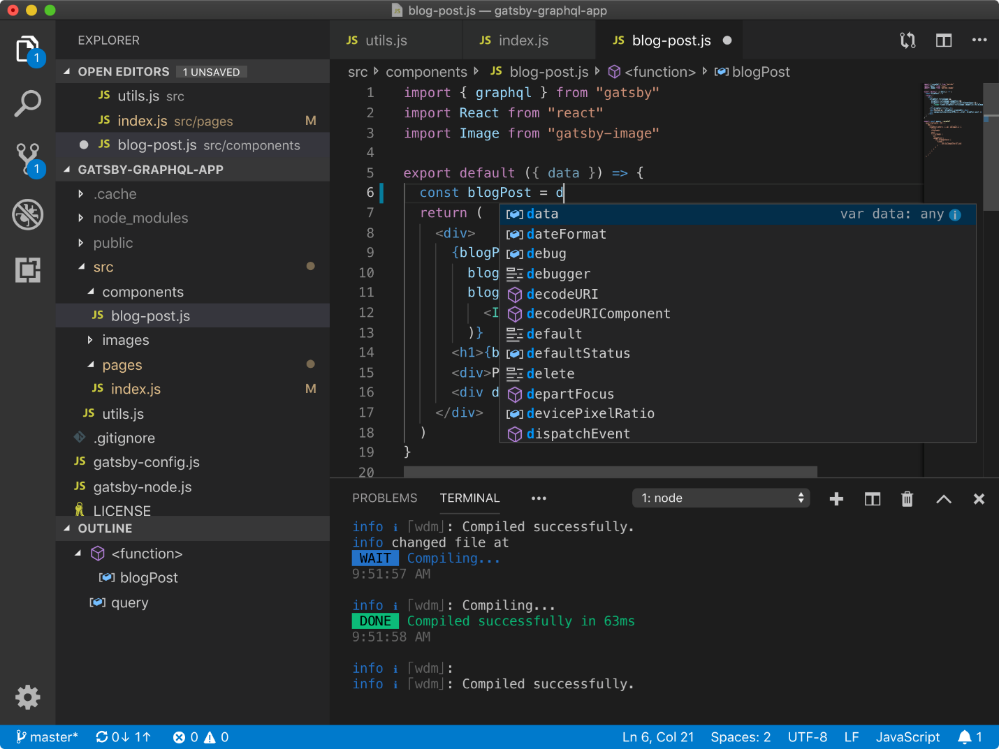


Figure 41 : Visual Studio Code

Visual Studio Code (VSCode) is a popular, lightweight, and powerful source code editor. It offers numerous benefits coding projects, making it a top choice for developers across various programming languages and platforms.

1. **Cross-platform compatibility**

major operating systems, Windows, macOS, and Linux,

1. **Extensibility**

extensions and plugins, allowing developers to customize

1. **IntelliSense**

VSCode offers advfeaturesode completion, navigation, and understanding features through its IntelliSense feature.

1. **Git integration**

The built-in Git support in VSCode simplifies version control, making it easier for developers to track changes, collaborate, and manage their codebase.

1. **Debugging capabilities**

VSCode offers a powerful built-in debugger, which supports various programming languages and platforms.

1. **Customizable UI**

The editor's interface is highly customizable, allowing developers to tweak the layout, color scheme, and more to suit their preferences and improve their workflow.

1. **Integrated terminal**

VSCode comes with an integrated terminal that allows developers to execute commands, run scripts, and interact with their development environment without leaving the editor.

1. **Language support**

The editor supports a vast array of programming languages, making it suitable for many different types of coding projects.

1. **Collaboration features**

The Live Share extension enables real-time collaboration between developers, allowing them to share their workspace and code together, regardless of their physical location.

1. **Active community and support**

Visual Studio Code has a large and active community, which contributes to the development of the editor and its extensions.

In summary, Visual Studio Code provided a powerful, extensible, and user-friendly environment for developers working on various coding projects. Its wide array of features, support for multiple languages, and strong community backing make it a top choice for many programmers.

# PROPOSED SOLUTION

## Design

### Basic Functionality Brief:

* The website will allow users to browse through a range of elderly services such as basic internet courses, booking public office visits , medicine delivery and hospital visits.
* The users will be able to search for services based on their location and service requirements.
* Once the user selects a service, they will be able to view details about the service, such as the service provider's name, contact details, pricing, and availability.
* The user will then be able to book the service by selecting a preferred date and time and entering all their contact or booking details.
* Overall, the website's goal is to make it easy for elderly people and their caregivers to find and book high-quality services that meet their needs.

### Design Snippets and Images

Diagram

Description automatically generated

Figure 42 : Basic Architecture Diagram

Diagram

Description automatically generated

Figure 43 : Technology based Architecture

Logo

Description automatically generated

Figure 44 : Technologies Used

Diagram

Description automatically generated

Figure 45 : UML Class Diagram

This UML Diagram above gives a basic flow of class diagram involving the steps needed for the website. The flow and steps I plan during this project also showing the important roles and functions I will try to follow while coding the project

## Implementation

### Database (MySQL)

A picture containing text, screenshot, indoor

Description automatically generated

Figure 46 : PhpMyAdmin

Tables and Database scheme

Graphical user interface, application

Description automatically generated

Figure 47 : Database Tables

**Why MySQL ?**

* MySQL is suitable for applications that require complex data relationships, transactions, and querying capabilities, such as e-commerce platforms, content management systems, or business applications.
* MySQL is an open-source, standalone database system and can be hosted on-premises or in the cloud.
* It is not tied to a specific suite of services.
* MySQL is a relational database, which means it stores data in tables with predefined schemas.
* It is designed for structured data storage and supports transactions and complex queries

**Conclusion**

I wanted to use MySQL because my application has mixed requirements. Login and other user entries also use MySQL for structured data storage and complex querying.

### Frontend Technologies (Bootstrap and HTML CSS)



Figure 48 : Bootstrap

**Why Bootstrap ?**

* **Timesaving**

CSS framework with pre-designed components and styles, which can speed up the development process by reducing the time spent on writing CSS from scratch.

* **Responsive Design**

Bootstrap's grid system and responsive utilities make it easy Creating web designs that are responsive and capable of adjusting to various screen sizes and devices.

* **Consistent Look and Feel**

Using Bootstrap components ensures a consistent look and feel across your application, as well as compatibility with various browsers.

* **Customizable**  
  Bootstrap can be customized according to your design requirements, allowing you to change colors, fonts, and other styles while still leveraging its predefined components.
* **Large Community**

Bootstrap has a large and active community, which means you can find plenty of resources, plugins, and tutorials online to help you learn and use the framework effectively.

Text

Description automatically generated

Figure 49 : Bootstrap Layout

**Why HTML, CSS, and JavaScript ?**

* **Fundamental Technology**

HTML, CSS, and JS are the core technologies for building any web-based frontend. HTML is used for structuring content, CSS for styling, and JS for adding interactivity and dynamic content.

* **Browser Compatibility**

Modern browsers support HTML, CSS, and JS natively, ensuring your website or web application will work across various platforms without requiring additional plugins or extensions.

* **Wide Range of Libraries and Frameworks**

There are numerous libraries and frameworks available for HTML, CSS, and JS that can help you create complex and interactive frontend experiences. Examples include jQuery (JS library), React, Angular, and Vue.js (JS frameworks), and Sass or Less (CSS preprocessors).

* **Developer Familiarity**

Most web developers are already familiar with HTML, CSS, and JS, making it easier to find and onboard developers for your project.

* **SEO and Accessibility**

HTML, CSS, and JS, when used properly, can contribute to better search engine optimization (SEO) and accessibility for your website, making it more discoverable and user-friendly.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta content="width=device-width, initial-scale=1.0" name="viewport">

    <title>El-Care</title>

    <!-- Favicons -->

    <link href="assets/img/favicon-logo.png" rel="icon">

    <link href="assets/img/favicon-logo.png" rel="apple-touch-icon">

    <!-- Google Fonts -->

    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

    <!-- Vendor CSS Files -->

    <link href="assets/vendor/animate.css/animate.min.css" rel="stylesheet">

    <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">

    <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">

    <link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">

    <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">

    <link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

    <!-- Template Main CSS File -->

    <link href="assets/css/mystyle.css" rel="stylesheet">

</head>

<body>

    <!-- ======= Header ======= -->

    <header id="header" class="fixed-top d-flex align-items-center">

        <div class="container d-flex align-items-center">

            <h1 class="logo me-auto"><a href="index.php">El-Care</a></h1>

            <nav id="navbar" class="navbar">

                <ul>

                    <li><a href="about.php">About Us</a></li>

                    <li><a href="testimonials.php">Testimonials</a></li>

                    <li><a href="ourteam.php">Our Team</a></li>

                    <li><a href="services.php">Services Offered</a></li>

                    <li><a href="contact.php">Contact Us</a></li>

                </ul>

                <i class="bi bi-list mobile-nav-toggle"></i>

            </nav>

            <!-- .navbar -->

        </div>

    </header>

    <!-- End Header -->

    <main id="main">

        <!-- ======= Breadcrumbs ======= -->

        <section id="breadcrumbs" class="breadcrumbs">

            <div class="container">

                <div class="d-flex justify-content-between align-items-center">

                    <h2>Our Team Will Contact You</h2>

                    <ol>

                        <li><a href="index.php">Home</a></li>

                        <li>Thank You</li>

                    </ol>

                </div>

            </div>

        </section>

        <!-- End Breadcrumbs -->

        <!-- <h1>Thank you!</h1> -->

        Thank you for submitting the form. We will contact you soon with a confirmation of the Booking!

    </main>

    <!-- End #main -->

</body>

<a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Vendor JS Files -->

<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="assets/vendor/glightbox/js/glightbox.min.js"></script>

<script src="assets/vendor/isotope-layout/isotope.pkgd.min.js"></script>

<script src="assets/vendor/swiper/swiper-bundle.min.js"></script>

<script src="assets/vendor/waypoints/noframework.waypoints.js"></script>

<script src="assets/vendor/php-email-form/validate.js"></script>

<!-- Template Main JS File -->

<script src="assets/js/main.js"></script>

</html>

**Conclusion**

By choosing Bootstrap combined with HTML, CSS for my front-end development, I can create responsive, visually appealing, and interactive web applications while leveraging a well-established and widely supported technology stack.

### Backend Technologies

**Why PHP ?**

* **Mature Language**

PHP has been around for a long time and has a well-established ecosystem. It is widely used for web development and has a large community that can provide support and resources.

* **Easy to Learn**

PHP has a relatively low learning curve, making it accessible to developers with varying levels of experience.

* **Built for Web Development**

PHP was designed specifically for web development, and its syntax and features are tailored for building web applications efficiently.

* **Integration with Web Servers**

PHP can be easily integrated with popular web servers like Apache and Nginx, simplifying the deployment process.

* **Wide Range of Frameworks**

PHP offers various frameworks like Laravel, Symfony, and CodeIgniter, which can help you structure your code and speed up development.

* **Cost-effective Hosting**

PHP hosting is widely available and often more affordable than hosting for some other backend languages.

    <?php

    if (isset($\_POST['Signin'])) {

        $query = "SELECT \* FROM `admin-login` WHERE `Admin\_Name`='$\_POST[AdminName]' AND `Admin\_Password`='$\_POST[AdminPassword]'";

        $result = mysqli\_query($con, $query);

        if (mysqli\_num\_rows($result) == 1) {

            session\_start();

            $\_SESSION['AdminLoginId'] = $\_POST['AdminName'];

            header("location: display.php");

        } else {

            echo "<script>alert('Incorrect Password');</script>";

        }

    }

    ?>

<?php

include 'connect.php';

if (isset($\_POST['submit'])) {

  $name = $\_POST['name'];

  $email = $\_POST['email'];

  $phone = $\_POST['phone'];

  $address = $\_POST['address'];

  $bookservice = $\_POST['bookService'];

  $bookmember = $\_POST['bookMember'];

  $eventDT = $\_POST['eventdt'];

  $sql = "insert into `crud`(Name,Email,Phone,Address,bookService,bookMember,eventdt) values('$name','$email','$phone','$address','$bookservice','$bookmember','$eventDT')";

  $result = mysqli\_query($con, $sql);

  if ($result) {

    echo "Data inserted successfully";

    header('location:thankyou.php');

  } else {

    die(mysqli\_error($con));

  }

}

?>

**Why JavaScript ?**

* **Single Language for Frontend and Backend**

Using JavaScript with Node.js allows you to develop both the frontend and backend of your application using a single programming language, which can streamline development and make it easier to find developers.

* **Non-blocking, Asynchronous I/O**
* Node.js leverages an event-driven design and non-blocking I/O, enabling efficient management of multiple concurrent connections and the development of scalable, high-performance applications.
* **Rich Ecosystem**

JavaScript has a vast ecosystem with a wide range of libraries and frameworks available, which can help you build feature-rich applications quickly.

* **Real-time Applications**

real-time applications like chat applications, online gaming platforms, or live data streaming services due to its asynchronous nature and ability to handle real-time data efficiently.

* **Easy to Scale**

Node.js applications are easy to scale, both horizontally and vertically, which can be important if you expect your application to grow and serve many users.

  /\*\*

   \* Back to top button

   \*/

  let backtotop = select('.back-to-top')

  if (backtotop) {

    const toggleBacktotop = () => {

      if (window.scrollY > 100) {

        backtotop.classList.add('active')

      } else {

        backtotop.classList.remove('active')

      }

    }

    window.addEventListener('load', toggleBacktotop)

    onscroll(document, toggleBacktotop)

  }

**Conclusion**

By choosing PHP and JavaScript (Node.js) as backend technologies, you can take advantage of the strengths of both languages, such as PHP's maturity and web development focus, along with JavaScript's versatility and performance capabilities

### Cloud Service and Connections

**Why AWS ?**

* **Plan your Architecture**

Before starting, define your requirements and plan your infrastructure architecture accordingly. Consider factors like scalability, availability, security, and cost optimization.

* **Choose the right services**

AWS offers a vast array of services. Select the ones that align with your requirements. For example, EC2 for virtual servers, S3 for object storage, RDS for managed databases, Lambda for serverless computing, etc.

* **Security**

Implement proper security measures to protect your AWS resources. Use IAM to control access, encrypt sensitive data using AWS Key Management Service (KMS), and follow AWS security best practices.

* **Cost optimization**

AWS offers various pricing models, and costs can add up quickly if not managed properly. Monitor your resource usage, leverage tools like AWS Cost Explorer and AWS Trusted Advisor and consider utilizing services like EC2 Spot Instances or Reserved Instances for cost savings.

* **High availability and fault tolerance**

Leverage AWS services to design highly available and fault-tolerant systems. Use Elastic Load Balancers (ELB) for distributing traffic, configure Auto Scaling to handle variable workloads, and consider deploying across multiple Availability Zones (AZs) for redundancy.

* Automation and orchestration

AWS provides automation tools like AWS CloudFormation and AWS Elastic Beanstalk. Use Infrastructure as Code (IaC) to provision and manage your AWS resources consistently and consider utilizing AWS Lambda and AWS Step Functions for serverless workflows.

* **Monitoring and logging**

Set up monitoring and logging to gain insights into your AWS environment. Use services like Amazon CloudWatch for monitoring resource utilization and performance and enable logging for services to track and analyze events and errors.

* **Disaster recovery**

Implement a disaster recovery plan to ensure business continuity. Consider utilizing services like Amazon S3 for data backups, AWS Storage Gateway for hybrid cloud storage, and AWS Backup for centralized backup management.

* **Networking**

Configure your networking components appropriately. Use Amazon VPC to create isolated virtual networks, set up security groups and network access control lists (ACLs) for controlling traffic, and consider using AWS Direct Connect for dedicated network connectivity.

* **Documentation and Support**

Take advantage of the extensive AWS documentation, whitepapers, and best practice guides. Additionally, AWS offers support plans with varying levels of assistance, including access to the AWS Support Center and technical support from AWS experts.

Graphical user interface, text, application

Description automatically generated

Figure 50 : AWS Management Console Page

Graphical user interface, text, application

Description automatically generated

Figure 51 : Instance Created for the Website

Graphical user interface, application

Description automatically generated

Figure 52 : Putty Login with Instance Details

Graphical user interface, text, application

Description automatically generated

Figure 53 : Putty Login with Private Key

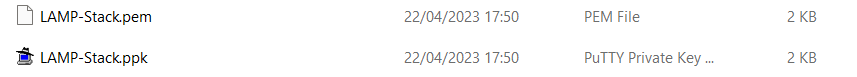


Figure 54 : .pem and .ppk files for the Instance key

Text

Description automatically generated

Figure 55 : Login as Ubuntu (Linux Instance)

INSTALL LAMP STACK ON AWS - UBUNTU 18

Step 1 — Installing Apache and Updating the Firewall

sudo apt update

sudo apt upgrade

sudo apt install apache2

sudo ufw app list

sudo ufw app info "Apache Full"

sudo ufw allow in "Apache Full"

APACHE INSTALLED SUCCESFULLY TILL HERE, YOU CAN CHECK BY ENTERING YOUR PUBLIC IP OR PUBLICK DNS ADDRESS - http://your\_server\_ip

==================================================

Step 2 — Installing MySQL

sudo apt install mysql-server

sudo mysql\_secure\_installation

sudo mysql

SELECT user,authentication\_string,plugin,host FROM mysql.user;

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password'; // please note here replace the "password" with yours.

FLUSH PRIVILEGES;

SELECT user,authentication\_string,plugin,host FROM mysql.user;

exit

At this point, your database system is now set up and you can move on to installing PHP, the final component of the LAMP stack.

=========================================================

Step 3 — Installing PHP

sudo apt install php libapache2-mod-php php-mysql

In most cases, you will want to modify the way that Apache serves files when a directory is requested. Currently, if a user requests a directory from the server, Apache will first look for a file called index.html. We want to tell the web server to prefer PHP files over others, so make Apache look for an index.php file first.

sudo nano /etc/apache2/mods-enabled/dir.conf

Move the PHP index file (highlighted above) to the first position after the DirectoryIndex specification, like this:

<IfModule mod\_dir.c>

    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

-----------------------------------

sudo systemctl restart apache2

sudo systemctl status apache2

Press Q to exit this status output.

Check PHP Version by entering

php -v

Install the commonly required php modules by using the below commands - do remeber replace the php version number with your by checking the php -v command

For Example if the php -v command shows 7.4 version installed then you have to replace the 7.2 with 7.4 in the below command

sudo apt install php7.2-common php7.2-mysql php7.2-xml php7.2-xmlrpc php7.2-curl php7.2-gd php7.2-imagick php7.2-cli php7.2-dev php7.2-imap php7.2-mbstring php7.2-opcache php7.2-soap php7.2-zip php7.2-intl -y

sudo systemctl restart apache2

==================================================

Step 4 — Testing PHP Processing on your Web Server

sudo nano /var/www/html/info.php

This will open a blank file. Add the following text, which is valid PHP code, inside the file:

<?php

phpinfo();

?>

The address you will want to visit is:

http://your\_ip/info.php

You will get the php info page

===========================

PHPMYADMIN INSTALL STEPS BELOW

Step 1 — Installing phpMyAdmin

sudo apt update

sudo apt install phpmyadmin php-mbstring php-gettext

Warning: When the prompt appears, “apache2” is highlighted, but not selected. If you do not hit SPACE to select Apache, the installer will not move the necessary files during installation. Hit SPACE, TAB, and then ENTER to select Apache.

sudo phpenmod mbstring

sudo systemctl restart apache2

===================================================

http://your\_domain\_or\_IP/phpmyadmin

====================================PERMISSIONS ADJUSTMENT================================

Step 2: Locate the PHP configuration file

Determining the right PHP configuration file can be very confusing especially because the ‘php.ini’ file can be located on a different folder depending on the PHP version.

The correct php.ini file should be in the Apache directory (e.g. ‘/etc/php/7.1/apache2/php.ini’). This will depend on the version of PHP. For instance, in Php7.2, the configuration file is located on ‘/etc/php/7.2/apache2/php.ini’

==============================================================

Step 3: Edit the Php Configuration file

sudo nano /etc/php/7.1/apache2/php.ini

Standard ‘php.ini’ settings file - Change the INI settings according to the below values:

memory\_limit = 128M

upload\_max\_filesize = 50M

post\_max\_size = 50M

max\_execution\_time = 120

sudo service apache2 restart

Step 4: Verify the php.ini settings

Refreshing the info.php page should now show your updated settings. Remember to remove the info.php when you are done changing your PHP configuration.

=========================================================================

Important Notes - Common Issues during/after PHP Install

MOST IMPORTANT

PERMISSION - YOU SHOULD OWN THE FILE BEFORE YOU CAN EDIT - YOU SHOULD KNOW THE USERNAME OF THE OPERATING SYSTEM.

=====================

Issues: Website pages not visible, not able to edit the files/folder , permsission denied issue, .htaccess not able to rewrite links.

Execute the Comands below to set proper file permissions on Directories and files.

sudo chown -R ubuntu:root /var/www/html

sudo find html -type d -exec chmod 775 {} \;

sudo find html -type f -exec chmod 664 {} \;

======================================================

Enabling mod\_rewrite on apache2

By default, Apache does not allow the use of ‘.htaccess’ file so you will need to edit the configuration of each website’s virtual host file by adding the following code:

OWN THE APACHE2 FOLDER FIRST IF YOU WANT TO EDIT VIA FILEZILLA OR FTP - sudo chown -R ubuntu:root /etc/apache2/ - Revert back to root:root chown when done.

OR VIA SSH TERMINAL

sudo nano /etc/apache2/apache2.conf

Change the setting as below : AllowOverride All

<Directory /var/www/html>

    Options Indexes FollowSymLinks MultiViews

    AllowOverride All

    Require all granted

</Directory>

------------

sudo a2enmod rewrite

sudo service apache2 restart

=================

TO VERIFY ANY CONFIG ERRORS OR MISTAKE OR SYSNTAX ERRORS IN APACHE CONFIG FILE, RUN THE BELOW COMMAND - Its should show Syntax OK , if it not then there is some error in config due to which apache will not start.

sudo service apache2 status (Press Q to exit)

# apachectl configtest

Syntax OK

# apachectl -t

Syntax OK

Commands to Install LAMP Stack and Upload Website on Instance

Graphical user interface, application

Description automatically generated

Figure 56 : Website Running on Instance

Graphical user interface, text, application

Description automatically generated

Figure 57 : DB running on Instance

## Testing /Evaluation

Testing and evaluation are critical components of developing a successful elderly service booking website. Here are some ways to test and evaluate the website:

* **Usability testing**

Conducting usability testing with elderly users can help ensure that the website is easy to use and understand. Test users can be asked to perform various tasks such as finding and booking a service, leaving a review, or navigating the website.

* **Accessibility testing**

Accessibility testing can ensure that the website is accessible to elderly users with different abilities. This testing can include using assistive technologies, such as screen readers or magnifiers, to test the website's functionality.

* **Performance testing**

Performance testing can ensure that the website can handle the expected traffic and load times. This testing can include stress testing, load testing, and performance monitoring.

* **Security testing**

Security testing can ensure that the website is secure and can protect user data. This testing can include penetration testing, vulnerability scanning, and security code reviews.

* **User feedback**

Collecting user feedback through surveys or feedback forms can help identify areas for improvement and provide insight into user needs and preferences.

* **Analytics and metrics**

Tracking website analytics and metrics, such as user engagement, conversion rates, and bounce rates, can provide insights into how users are interacting with the website and help identify areas for improvement.

Throughout the development of the website, continuous testing and evaluation should be conducted to ensure that it effectively caters to the needs and expectations of elderly users and their caregivers. This ongoing process is crucial for achieving a website that meets their requirements.

## 

## Demonstration Record

Graphical user interface, application

Description automatically generated

Figure 58 : Website Home Page

Graphical user interface, text, application, email

Description automatically generated

Figure 59 : About Us Page

Graphical user interface, text, application

Description automatically generated

Figure 60 : About Us Page Continued

Graphical user interface, application, Teams

Description automatically generated

Figure 61 : Testimonials Page

Graphical user interface, application, Teams

Description automatically generated

Figure 62 : Our Team Page

Graphical user interface, application

Description automatically generated

Figure 63 : Contact Us Page

A screenshot of a computer

Description automatically generated with medium confidence

Figure 64 : Contact Page Data Entry

A screenshot of a computer

Description automatically generated with medium confidence

Figure 65 : Form Submission Via Formspree

A screenshot of a computer

Description automatically generated with medium confidence

Figure 66 : Collecting Form Data in Formspree

Graphical user interface, application, Teams

Description automatically generated

Figure 67 : Services Page

Text

Description automatically generated

Figure 68 : Services Page Continued

Graphical user interface, text, application, email

Description automatically generated

Figure 69 : Booking Form Page

Graphical user interface, text, application, email

Description automatically generated

Figure 70 : Booking Confirmation Page

Graphical user interface, application, website

Description automatically generated

Figure 71 : Admin Login Page

Graphical user interface, website

Description automatically generated

Figure 72 : User Registration Page

Graphical user interface, application, website

Description automatically generated

Figure 73 : User Login Page

Graphical user interface, application, website

Description automatically generated

Figure 74 : Employee Login Page

Graphical user interface, application

Description automatically generated

Figure 75 : Employee Login Display List

Graphical user interface, application

Description automatically generated

Figure 76 : Admin Login Display List

# CONCLUSIONS

## original aims and objectives of the project

As a student working on an elderly service booking website, there were several reasons why I might not have been able to meet the original aims and objectives.

* **Limited expertise and experience**

As a student, I have faced limitations in terms of my technical skills, knowledge, and experience in website development and project management. Building a fully functional and user-friendly website with complex features requires expertise in various areas such as coding, design, and usability testing. If I lacked the necessary expertise or guidance, it could have hindered my ability to achieve the original aims and objectives.

* **Time constraints and competing priorities**

Balancing academic commitments, personal obligations, and extracurricular activities can be challenging. I had limited time to dedicate to the project or had to juggle multiple responsibilities simultaneously, it impacted my progress. Insufficient time and attention to the project can result in delays, rushed work, or an inability to fully implement the intended features, thus affecting the achievement of the original aims and objectives.

* **Limited resources and support**

As a student, I encountered resource constraints, such as limited access to funding, software tools, or technical support. Lack of financial resources could have limited your ability to invest in necessary components, such as domain registration or hosting. Additionally, limited access to mentorship or guidance from experienced professionals could have made it difficult to overcome technical challenges or make informed decisions, impacting the overall success of the project.

* **User feedback and testing limitations**

Developing a user-centered website requires gathering feedback and conducting usability testing with the target audience. As a student, I faced challenges in recruiting a diverse group of elderly users or lacked the resources to conduct comprehensive user testing. Insufficient user feedback can result in design or functionality gaps that prevent the website from meeting the needs and expectations of its intended users.

* **Evolving project scope and requirements**

Throughout the course of the project, I encountered changes in the project's scope or requirements. These changes could have been due to evolving user view, feedback from supervisors, or a deeper understanding of the problem space. Adapting to these changes within the constraints of time and resources can be demanding, and it might have led to deviations from the original aims and objectives.

It's important to recognize that as a student, I faced unique challenges when working on a project like an elderly service booking website. Reflecting on these challenges can provide valuable insights into areas for improvement and help inform future projects or initiatives. Remembering, that learning from setbacks is an essential part of the learning experience and can contribute to my growth as a student and aspiring professional.

## Experience of undertaking an individual project

Undertaking an individual project was a rewarding and challenging experience. It provides an opportunity to take ownership of a project from start to finish, allowing you to apply your skills, knowledge, and creativity in a focused and independent manner. I learnt the following during this year

* **Project Selection**
* **Defining Objectives**
* **Planning and Scheduling**
* **Research and Preparation**
* **Execution and Implementation**
* **Problem-solving and Iteration**
* **Documentation**
* **Evaluation and Reflection**
* **Presentation and Communication**
* **Learning and Growth**

## Evidence that student understands and is aware of the limitations of the work conducted and would be able to develop it further in future work

* To establish partnerships with reputable service providers specializing in healthcare, personal care, home maintenance, leisure activities, and transportation, to offer a diverse and comprehensive range of services for the elderly users.
* To promote the elderly service booking website through targeted marketing campaigns, community outreach, and collaboration with relevant organizations, ensuring widespread awareness and adoption of the platform among the target demographic.
* To evaluate the impact of the platform on the quality of life and independence of elderly users, by tracking key performance indicators such as user satisfaction, frequency of service bookings, and reduction in barriers to access essential services.
* To create a supportive online community within the platform, where elderly users can engage with each other, share experiences, and receive peer-to-peer support, thus fostering social interaction and reducing feelings of isolation.
* The website will also have a review system that allows users to leave feedback about the service they received. This will help other users make informed decisions when selecting a service provider.
* To ensure the safety of the elderly, the website will verify the service providers' credentials and conduct background checks on them.
* The website will also have a system in place to handle complaints and resolve disputes between service providers and users.
* To include payment system in future with Stripe or PayPal or Gpay
* To see the business aspect and considering following a App Development Approach as well

# APPENDICES

For the convenience of sharing the source code and related project files, the GitHub repository link is provided below:

GitHub Repository: **https://github.com/AmritaGiri/Final-Year-Project-Elcare**

By accessing the above link, you can explore the complete source code, documentation, and any other relevant files associated with the project. The repository serves as a centralized location where you can view, download, and collaborate on the project.

Please note that the GitHub repository will allow you to track any updates, revisions, or contributions made to the project over time. Additionally, it provides an avenue for version control, issue tracking, and collaboration with other developers or stakeholders.

Feel free to access the repository and utilize the resources available to gain a deeper understanding of the project and its implementation. If you need clarification or additional help, feel free to contact us without hesitation.

Name : Amrita Giri

Email : [D00226038@student.dkit.ie](mailto:D00226038@student.dkit.ie)

LinkedIn : **https://www.linkedin.com/in/amritagiri/**

# REFERENCES

1. Verzeo Blogs. 2022. Which is the Best Operating System?? iOS vs Android vs Windows. [ONLINE] Available at: https://blog.verzeo.com/which-is-the-best-operating-system/#:~:text=Android%20is%20open%20source%2C%20while%20iOS%20and%20Windows,Apple%20Store%2C%20and%20Android%20has%20Google%20Play%20Store.. [Accessed 22 November 2022].
2. What is Cloud Computing? | IBM. 2022. What is Cloud Computing? | IBM. [ONLINE] Available at: https://www.ibm.com/cloud/learn/cloud-computing. [Accessed 22 November 2022].
3. Wikipedia. 2022. Amazon Web Services - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Amazon\_Web\_Services. [Accessed 24 November 2022].
4. Wikipedia. 2022. Microsoft Azure - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Microsoft\_Azure. [Accessed 24 November 2022].
5. WebsiteBuilderPoint |. 2022. What Is Client-Side in Web Development?. [ONLINE] Available at: https://www.websitebuilderpoint.net/what-is-client-side-in-web-development//. [Accessed 24 November 2022].
6. WebsiteBuilderPoint |. 2022. What Is Server-Side Development in Web Technology?. [ONLINE] Available at: https://www.websitebuilderpoint.net/what-is-server-side-development-in-web-technology/#:~:text=Server-side%20development%20is%20a%20process%20where%20the%20code,when%20the%20interface%20is%20not%20accessible%20to%20users.. [Accessed 24 November 2022].
7. Useful apps and websites for older people | Handicare News. 2022. Useful apps and websites for older people | Handicare News. [ONLINE] Available at: https://www.ageukmobility.co.uk/mobility-news/article/useful-apps-and-websites-for-older-people. [Accessed 24 November 2022].
8. Goodnet. 2022. 7 of the Best Apps for People with Disabilities - Goodnet. [ONLINE] Available at: https://www.goodnet.org/articles/7-best-apps-for-people-disabilities-list. [Accessed 24 November 2022].
9. Tunstall. 2017. 9 apps and websites for AT and disability services that could change your life - tunstallhealthcare. [ONLINE] Available at: https://blog.tunstallhealthcare.com.au/disability/9-apps-and-websites-for-at-and-disability-services-that-could-change-your-life/. [Accessed 28 November 2022].
10. Educba. 2022. Google Cloud Platform. [ONLINE] Available at: https://www.educba.com/google-cloud-platform/. [Accessed 28 November 2022].
11. Educba. 2022. GCP vs AWS vs Azure. [ONLINE] Available at: https://www.educba.com/gcp-vs-aws-vs-azure/. [Accessed 28 November 2022].
12. Focaloid Technologies. 2021. Types of Application Development Services. [ONLINE] Available at: https://www.focaloid.com/blog/types-of-application-development-services/#:~:text=Types%20of%20application%20development%20include%20desktop%20application%20development%2C,full%20form%20of%20API%20is%20Application%20Program%20Interface. [Accessed 28 November 2022].
13. Wikipedia. 2022. Firebase - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Firebase. [Accessed 28 November 2022].
14. Olga Anoshyna. 2022. Why Choose Firebase as Cloud Backend for your Mobile Apps - Super Dev Resources. [ONLINE] Available at: https://superdevresources.com/why-use-firebase/. [Accessed 28 November 2022].
15. Willvick. 2022. Advantages of Using Bootstrap - Willvick. [ONLINE] Available at: https://willvick.com/advantages-of-using-bootstrap/#:~:text=Advantages%20of%20Using%20Bootstrap%201%20Open%20Source%20%E2%80%94,Bootstrap%20predefined%20design%20templates%20and%20classes.%20More%20items. [Accessed 28 November 2022].
16. Upwork. 2021. A Beginner's Guide to Back-End Development. [ONLINE] Available at: https://www.upwork.com/resources/beginners-guide-back-end-development. [Accessed 28 November 2022].
17. Upwork. 2021. Server-Side Scripting: Back-End Web Development Technology. [ONLINE] Available at: https://www.upwork.com/resources/server-side-scripting-back-end-web-development-technology. [Accessed 28 November 2022].
18. Jigsaw Academy. 2022. Top 15 IAAS Examples You Need To Know In 2021. [ONLINE] Available at: https://www.jigsawacademy.com/blogs/cloud-computing/iaas-examples/. [Accessed 29 November 2022].
19. Jigsaw Academy. 2022. Top 10 SAAS Examples. [ONLINE] Available at: https://www.jigsawacademy.com/blogs/cloud-computing/saas-examples/. [Accessed 29 November 2022].
20. Serverless Works. 2022. Serverless Computing Examples Explained | Serverless Works. [ONLINE] Available at: https://serverlessworks.com/examples#:~:text=Some%20of%20the%20examples%20of%20core%20serverless%20technologies,5%20Cloud%20functions%206%20Cloud%20pub%2Fsub%207%20Cloud. [Accessed 29 November 2022].
21. YourTechDiet. 2022. Why opt for FAAs? Here are 4 Real Life Examples. [ONLINE] Available at: https://yourtechdiet.com/blogs/function-as-a-service-faas-examples/. [Accessed 29 November 2022].
22. Salesforce.com. 2023. 12 Benefits of Cloud Computing and Its Advantages - Salesforce.com. [ONLINE] Available at: https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/. [Accessed 23 January 2023].
23. Clutch.com. 2015. Mobile App Platforms: Hybrid, Native, Mobile Web - Clutch.com. [ONLINE] Available at: https://clutch.co/app-developers/resources/mobile-app-platforms-hybrid-native-mobile-web#:~:text=When%20building%20a%20mobile%20app%2C%20there%20are%20three,of%20each%20platform%20type%20and%20possible%20use%20cases.. [Accessed 23 January 2023].
24. Data Flair. 2022. Pros and Cons of JavaScript- data-flair.training.com. [ONLINE] Available at: https://data-flair.training/blogs/advantages-disadvantages-javascript/#:~:text=Advantages%20of%20JavaScript%201%201.%20Speed%20Since%20JavaScript,Functionality%20...%208%208.%20Versatility%20...%20More%20items. [Accessed 23 January 2023].
25. Sagar Sharma. 2023. 15 Top Front-End Technologies To Rule The Market. [ONLINE] Available at: https://credencys.com/blog/front-end-technologies/#:~:text=In%20simple%20words%2C%20it%20is%20a%20set%20of,up%20a%20website%2C%20web%20application%2C%20or%20mobile%20app.. [Accessed 27 January 2023].
26. Microsoft. 2023. What is IaaS?. [ONLINE] Available at: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-iaas/ [Accessed 04 February 2023].
27. Managed cloud services (2023) RapidScale. Available at: https://rapidscale.net/ (Accessed: 09 May 2023).
28. InformationWeek, serving the information needs of the Business Technology Community (no date) InformationWeek. Available at: https://www.informationweek.com/ (Accessed: 09 May 2023).
29. (No date) Formspree. Available at: https://formspree.io/forms/mleklwkd/integration (Accessed: 12 May 2023).
30. Mark Otto, J.T. (no date) Introduction, · Bootstrap. Available at: https://getbootstrap.com/docs/4.0/getting-started/introduction/ (Accessed: 12 May 2023).