The University of Manchester
Department of Computer Science
Project Report 2024

Automatic accessibility evaluation and analytics of NHS General Practices' websites

Author: Ruoqing Zheng

Supervisor: Dr. Markel Vigo

Abstract

Automatic accessibility evaluation and analytics
of NHS General Practices' websites

Author: Ruoqing Zheng

The importance of web accessibility in providing equal access to essential healthcare resources and services cannot be overlooked. If website information offered by healthcare services is not accessible, many individuals get excluded from it. As the number of people with disabilities seeking healthcare services continues to rise, it becomes critical to assess the accessibility of healthcare websites to ensure inclusivity. Previous studies have evaluated web accessibility using various tools and techniques. This emphasises the necessity of thorough assessments to identify and address accessibility barriers effectively. However, research has paid limited attention to healthcare platforms, particularly the National Health Service (NHS) General Practices' (GP) websites. The objective of this project is to use AChecker to systematically evaluate the web accessibility across all NHS GP websites in England. Also, the evaluation results will be regularly analysed and visualised on a dashboard. The accessibility analysis reveals that a significant proportion of NHS GP websites (60.8%) fail to meet even the minimum level of web accessibility conformance. This shows the urgent need for ongoing monitoring and improvement efforts to improve accessibility and ensure access to healthcare resources for all individuals. The evaluation of the dashboard usability achieves a score of 82.25 on the System Usability Scale, demonstrating a high level of user satisfaction and effectiveness.

Supervisor: Dr. Markel Vigo

Acknowledgements

I am very grateful to my supervisor, Dr. Markel Vigo, who has always patiently provided guidance and suggestions to me during my last year. I would not be able to complete my project without his help.

I would also like to thank my family and friends who have supported me during this stressful year. Your support and encouragement motivated me to carry on through each overwhelming night.

I appreciate all the participants who took part in the evaluation stage of my project. Your valuable feedback and praise also contributed to the success of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The World Wide Web is designed to offer a variety of services to users around the world. Its main advantage is its universality [26], which guarantees that everyone can access it, regardless of their locality, language, technique expertise, age, or the specification of their software and hardware devices. In particular, it extends universal accessibility to people with various types of mental or physical disabilities, including cognitive disability, visual impairment and hearing loss. In addition, improved accessibility also benefits non-disabled people in a variety of situations, such as older people, users with small screen devices, non-native speakers, and users with limited bandwidth connections. With the high-level accessibility web services, people with such disabilities are able to communicate and interact with the services more easily via alternative provided approaches.

To monitor web accessibility, researchers have paid increasing attention to web accessibility evaluation. One of the most discussed guidelines is the Web Content Accessibility Guidelines 2.0 (WCAG 2.0) developed by The World Wide Web Consortium (W3C) [26]. These guidelines provide a framework to ensure the web services are accessible to all people. A prevalent method to assess web accessibility is automated testing, which provides an affordable and time-saving solution. There is an increasing number of automatic evaluation tools available to developers and societies, such as WAVE, Siteimprove, AChecker and Google Lighthouse. These tools can be classified based on diverse criteria: cost (free or commercial), platform (local or online) and format of their reporting (with a wide range of output formats varying from PDF to HTML file). While automatic testing can efficiently help identify potential accessibility problems and can be integrated into all stages of web development, it may generate misleading results and may not report all existing issues. Hence it can only aid in evaluation, rather than substitute manual check [25]. Additional evaluation approaches are also necessary depending on the specific purposes and the expected outcomes. Manual inspection is usually conducted by a group of evaluators with specialised expertise, while user testing involves evaluation conducted by users with disabilities [1]. Both of these methods have strengths and weaknesses. They tend to be more reliable than automatic testing, since experts produce high-quality reports and users provide insights into a real-world situation. However they are both resource-intensive, requiring substantial labor and financial resources. Furthermore, even with guidelines provided, human judgement is subjective, it may not always reach a high degree of agreement.

Therefore, the evaluation of web accessibility is a considerably challenging task. Research has shown that accessibility guidelines and suggestions are actually adopted at a significantly low level [15], in spite of the abundance of evaluation services that are offered. This may be attributed to various factors. Among website developers, one of the primary obstacles is the lack of familiarity and awareness of taking accessibility into consideration at an early stage of development. Additionally, external reasons such as client attitudes towards web accessibility and the challenges involved in enforcing accessibility regulations further worsen the problem [23]. Also, the presence of ambiguity and limited reliability in the existing guidelines causes another barrier to their widespread adoption [9].

However, conducting accessibility evaluation and aiming to improve web accessibility continuously are still essential tasks which cannot be omitted to ensure web accessibility for all people. Nowadays, web accessibility has become more and more important in some of the online services that may be encountered in people's daily lives, including education and finance, especially in healthcare. The inaccessibility of health resources, for example, online general practices and hospitals, may lead to difficulty in appointment scheduling, restricted access to medical information, and barriers to telemedicine. All these obstacles have the potential to negatively impact the health outcomes of a wide range of individuals.

1.2 Aims and Outcomes

This work mainly focuses on the evaluation of web accessibility of all National Health Service (NHS) General Practices' (GP) websites in England. The goals include the following:

1. Build crawlers to systematically collect GP information, including GP names, postcodes, locations and URLs.
2. Embed the evaluation tool AChecker to assess web accessibility for each GP website.
3. Perform statistical analysis on the evaluation results to obtain trends and insights.
4. Create a dashboard to visualise the analysis findings, and update regularly update these results to reflect the latest results.
5. Analyse and update the evaluation results quarterly, ensuring that all findings are up-to-date and stored in a database. The stored results include the specific issue IDs, number of issues, different types of issues, and the timestamp recording the starting time of the evaluation process.

Originally, the plan was to build a single crawler to gather all relevant GP information directly from NHS websites. However, since NHS websites did not provide working URLs for each GP website, this task became significantly more complex. Therefore, the data collection process had to be divided among three separate crawlers, each tailored to gather specific data effectively.

To achieve these objectives, two main tasks are undertaken. The initial task involves the collection of data, including location information, GP names, and their official website Uniform Resource Locators (URLs) utilising Python Scrapy framework and Selenium, with storage of the acquired data in the MySQL database system. The second task is to establish a dashboard website using Django. In the backend, Application Programming Interfaces (APIs) including an evaluation tool AChecker and Postcodes.io are used for accessibility evaluation and location finding. The key information extracted from the reports generated by AChecker is then stored and analysed. In the frontend, ECharts is mainly adopted for interactive data visualisation, while Bootstrap is employed to build certain components of the website. Additionally, a scheduler utilising Django APScheduler library is implemented to perform the data analysis task in a regular fashion. In the end, the website is deployed on the PythonAnywhere platform, enabling external users to access and conduct usability evaluations.

The dashboard provides visualisation of the accessibility results from three distinct perspectives: overall, regional and individual GP website. In order to enable personalised exploration of accessibility results, a search bar is integrated, allowing users to retrieve the analysis results for specific GPs, by inputting the GP name, location, or postcode.

The evaluation process is divided into two parts. One is the code test, including both the unit and functional tests for the Django dashboard. The other is the usability evaluation, which involves a group of users to test manually and complete a user experience questionnaire, providing insights into their perception and interaction with the dashboard.

1.3 Report Structure

This report contains five chapters:

- Chapter 1 provides an overview of web accessibility, accessibility evaluation methodologies, and an introduction to the project.
- Chapter 2 discusses the WCAG 2.0 guidelines and some evaluation tools including AChecker, along with a comprehensive review of related works in the field.
- Chapter 3 describes the detailed methodology used in the development process.
- Chapter 4 discusses all the findings of the analysis and the results of the evaluation process.
- Chapter 5 gives a project conclusion, and reflects on the project limitations and possible improvements in the future.

Chapter 2

Background

This chapter explores the relevant work carried out by researchers and developers in this field, and discusses its methods and findings. Moreover, it introduces WCAG 2.0 and some existing web accessibility evaluation tools including AChecker. It also presents some limitations and gaps in the existing research.

2.1 Related Work

In recent years, the focus on web accessibility evaluation has increased significantly. Researchers have assessed the accessibility of the websites across various domains and countries. Akgül and Vatansever conducted a detailed evaluation of 25 government online resources in Turkey based on both WCAG 1.0 and 2.0 as the standards [5]. They adopted a mix of open-source and commercial tools such as TAW, Total Validator and WAVE to access these sites. Their findings revealed a concerning trend: none of the Turkish government websites fully addressed accessibility issues, and most failed to meet even the minimum web accessibility standards. Similarly, Acosta-Vargas et al. evaluated the websites of twenty of the world's most prestigious universities [4]. Using tools such as TAW and Examinator, they assessed these websites against the criteria outlined in WCAG 2.0. They adopted a methodology to help evaluators to understand and improve the guidelines. This method included sampling the target websites and analysing the overall accessibility level according to the sampling results. Their results showed that most of the university websites violated the guidelines, creating barriers for a large percentage of users.

Many researchers have also paid attention to the accessibility evaluation of healthcare websites. Sarita et al. [19] used evaluation tools such as AChecker and WAVE to assess the top six healthcare websites in India. They quantified the number of problems in each category and qualitatively analysed the accessibility level based on a Five-Level Accessibility Criteria they proposed. They divided the accessibility results into five levels: maximal, high, moderate, low, and minimal, each corresponding to a range of error percentages from low to high. The results indicated that all six sites had moderate to maximal accessibility. Furthermore, Sik-Lanyi and Mihálykó [22] extended a similar assessment to healthcare websites in nine European countries. Expert inspections were also involved in order to improve the evaluation process. They grouped the websites into Western-Northern and Eastern regions and compared the results. Also, they explored various perspectives such as website size and economic conditions, although no clear correlation was observed.

Although automated evaluation tools offer efficiency and convenience, their accuracy and coverage may be limited. In addition to automatic tools, researchers have proposed manual evaluation methods. The Barrier Walkthrough Method developed by Yesilada et al. [30], was one manual technique. This method was able to better identify obstacles and difficulties faced by users. In this approach, experts manually detected barriers as any condition that prevented users from achieving their goals while browsing a site. They considered various aspects, including users' goals, page characteristics, and the resulting effects. For each user category, a list of potential barriers was prepared for evaluation against web pages. By evaluating the impact of each potential scenario, the severity of each identified barrier could be determined.

Acosta-Vargas et al. used a more comprehensive evaluation approach [3]. The strategy yielded a better evaluation result by integrating heuristic evaluation, manual verification and automated tools. Following initial evaluations by automated tools, a manual review was conducted by domain experts. A questionnaire based on the Barrier Walkthrough method was used in the last stage. This combined method provided more insights into the evaluation and assessed the websites more reliably.

Furthermore, several studies focused on the analysis and evaluation of the results produced by the automated tools. Vigo et al. [24]. carried out a thorough investigation of the automated evaluation tools. They quantitatively evaluated the performance of six commonly used evaluation tools and analysed the completeness, coverage and correctness of each evaluation report. The findings demonstrated that most tools had limited coverage and completeness. This emphasised the importance of human judgment and the need to avoid solely depending on automated tests.

Similarly, Abduganiev et al. accessed and compared eight popular automated evaluation tools in terms of WCAG 2.0 conformance [2]. Various metrics were used to evaluate the performance of each tool, including: specificity, inter-reliability, intra-reliability, validity, efficiency and capacity. With different evaluation metrics, their findings were consistent with the results of Vigo's study, indicating that dependence on a single automated tool may be inadequate due to certain success criteria that could not be effectively automated. Instead, a strategic combination of multiple automated tools was found to enhance the overall quality of the evaluation results.

Additionally, there have been developments of web accessibility evaluation dashboards. Ramiro et al. built the "AccessWeb Barometer" software platform [18]. The platform was designed to evaluate multiple websites simultaneously, analysing and publishing the results according to the accessibility guidelines. It also allowed users to create the custom dashboards and analysis. The platform consists of a website layer, an analytic layer and a diagnostic layer. The analytics layer and the website layer conducted analysis and visualised the results on the dashboard, while the diagnostic layer provided users with options to review and check the issues manually. In the statistical analysis process, they analysed 1000 Portuguese company websites, validated and posted the results on the platform.

2.2 WCAG 2.0

The Web Content Accessibility Guidelines standard includes three versions: 2.0, 2.1 and 2.2 [29]. The W3C Web Accessibility Initiative (WAI) has consistently refined this accessibility standard, aiming to improve its clarity and testability. Additionally, WCAG 3 [27] has been developed and is currently in the exploratory stage. While more success criteria are provided in later versions, this report primarily focuses on WCAG 2.0. Due to its continued widespread adoption across various evaluation tools, it is selected as the standard guideline for this project.

The WCAG 2.0 guidelines provide a technical standard for more accessible web content, including natural content such as media (audio or videos), images, and text, and the code that presents the relationship and representation of the information. All criteria are divided into four principles [26]:

- **Perceivable:** It examines whether the content can be perceived easily without losing any structure or information.
- **Operable:** It examines whether the content can be interacted with a keyboard or other alternative devices, with enough time and clear guidance provided.
- **Understandable:** It examines whether the content can be read and understood, and recommends creating a predictable and error-proof interface.
- **Robust:** It examines whether the content is accessible to various users and devices.

Each guideline within the WCAG 2.0 provides multiple testable success criteria. These criteria are categorised into three levels: A, AA and AAA, also known as levels of conformance [28]. The three levels are defined as follows:

- **Level A conformance:** The page meets all A-level success criteria.
- **Level AA conformance:** The page meets all A-level and AA-level success criteria.
- **Level AAA conformance:** The page meets all three levels of success criteria.

This categorisation signifies the degree to which the web content satisfies specific requirements, and does not violate any (or all) of the success criteria. While Level A denotes the minimum level of conformance, striving towards a higher level is encouraged to enhance overall accessibility and usability.

2.3 Existing Evaluation Tools

There are a large number of evaluation tools with various types and features. Table 2.1 below shows some commonly used evaluation tools:

Table 2.1: A table of web accessibility evaluation tools.

Tools	Type	Paid or Free
Total Validator	browser plugin, desktop application, CI (command-line)	partially free
TAW	online website	free
Siteimprove	online website, browser plugin, API	partially free
Luma	browser plugin	free
IBM Equal Access	browser plugin, CI	free
Axe Monitor	online website, browser plugin	commercial
Wave	online website, browser plugin, API	partially free

The selection of evaluation tools for our system requires careful consideration of their functionality and compatibility with automated processes. Several tools, including Siteimprove and

Wave, offer APIs that could potentially be integrated into our system. However, these tools typically limit the number of free API requests (usually between 100 to 300 requests per account), with additional charges applying for extra requests. Given the scale of our project, which involves assessing approximately 6,000 URLs, these tools are not economically feasible due to the expenses in exceeding the request limits.

After evaluating various options, AChecker, which provides completely free API service with exhaustive evaluation reports, was finally adopted in our system. It is developed by Cantan Group, offering both online services and an API for evaluating the accessibility of a web page. It can assess the web page content by entering a URL, inputting HTML source code or uploading an HTML file [11]. It provides various guidelines including WCAG 2.0, categorised into conformance levels of A, AA, and AAA.

Through this API, AChecker generates evaluation reports in either HTML or XML format. The tool identifies three types of issues:

- **Known problems (errors):** These issues are identified with certainty.
- **Likely problems:** These are problems that are classified as probable barriers, and may require human judgement for further evaluation and resolution.
- **Potential problems:** These issues are beyond AChecker's current capability to identify, requiring manual checks to determine their existence and severity.

For every problem identified, AChecker offers detailed information, including the specific piece of source code that causes the issue, along with its precise location indicated by line and column numbers within the HTML file. Furthermore, the tool examines several HTML elements for each success criterion, and provides comprehensive insights for each identified issue:

1. A unique ID of each accessibility issue.
2. A short description of the issue.
3. A specific success criterion from WCAG 2.0 it violates.
4. A requirement that the element is expected to meet, in order to align with the criterion.
5. Code examples demonstrating both passing and failing instances of the test, aiding users to understand how the requirement is achieved in practice.

2.4 Limitation and Gaps

From the review of previous research in the field of web accessibility evaluation, it is evident that significant progress has been made. However, certain gaps and limitations still exist in previous literature. Although many studies have focused on assessing the accessibility across diverse domains, there is relatively limited research evaluating the accessibility of healthcare websites, particularly those related to NHS GP services in the UK. Moreover, the number of evaluated websites in these studies is often limited, potentially hindering the generalisation of their findings.

Furthermore, existing research mainly concentrates on conducting overall assessments of accessibility levels for individual websites, and therefore ignores other potential findings that could be obtained by examining from various angles, such as regional considerations. Also, several analyses are conducted as one-time assessments, which may cause the results to become obsolete or invalid as website content evolves over time. Additionally, to the best of my knowledge, the visualisation and presentation of the analysis results in some studies are inadequate, lacking sufficient clarity and interactivity.

To address these limitations, the current project aims to evaluate and improve the existing practices. Specifically, the project will conduct an exhaustive evaluation of the web accessibility of all NHS GP websites across England. Evaluation reports produced by the assessment tool will be analysed from three distinct perspectives, including overall, regional, and individual GP analyses. Importantly, the project will implement regular updates and assessment to ensure that the analysis results reflect the most current accessibility status of the websites. This also enables trend analysis over time.

In terms of visualisation, various interactive charts will be employed to improve user understanding and engagement. The dashboard will display tables and charts to give a general overview of web accessibility in England. Additionally, the maps will provide information about the accessibility levels in different regions. To help users understand the evaluation results in more detail, individual GP reports for each website will also be included. Through these improvements, the project aims to offer a more thorough and user-friendly approach to web accessibility assessment.

Chapter 3

Methodology

In this chapter, a step-by-step methodology and a workflow are provided. Subsequently, the architecture of both data collection and data visualisation processes is introduced. In the end, various testing methods are included for the project evaluation.

3.1 Workflow

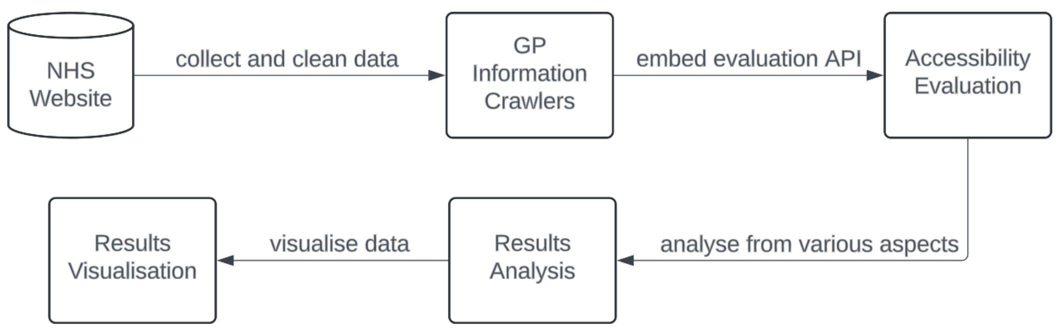

Figure 3.1: The workflow of the project.

Figure 3.1 illustrates the data flow process in the project, starting with the collection of GP information using multiple crawlers. The information includes GP names, postcodes, locations and URLs. Subsequently, the collected data undergoes data cleaning and is stored in the database. The homepage URLs of GP websites are then extracted and fed into the web accessibility API, AChecker, which generates accessibility reports for each GP website. From these reports, information such as the number of issues and unique issue IDs are then extracted and stored into the database. Utilising this data, various analyses from overall, regional and individual website aspects are conducted. Finally, the analysis results are visualised on the dashboard, providing the insights into web accessibility across NHS GP websites.

3.2 Architecture

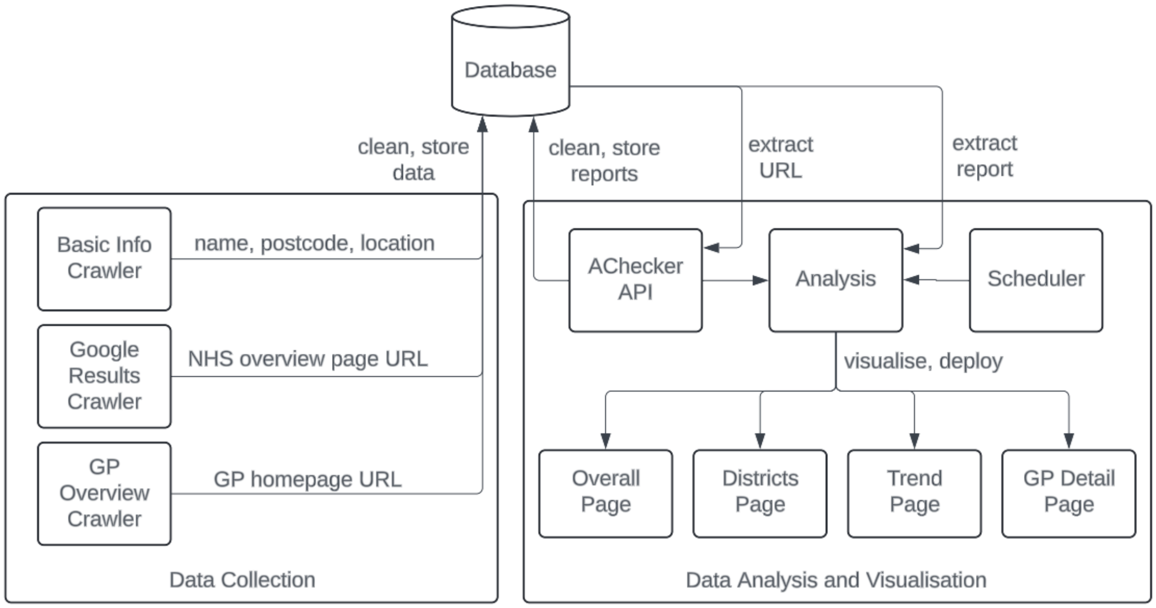

Figure 3.2: The overview of the project architecture.

Figure 3.2 shows the architecture of this project, which can be divided into two sub-projects: the crawling project and the analysis project. The crawling project collects and cleans GP data from different sources, and the analysis project is responsible for the analysis and visualisation of the accessibility evaluation results.

3.3 Crawling Project

The crawling project is responsible for collecting relevant GP information using Scrapy and Selenium. However, during the data collection phase, temporary link failures led to the inaccessibility of the overview pages for all GPs. Consequently, GP URLs could not be obtained using a single crawler and required additional search efforts through the Google search engine. As a result, the data collection process was divided among three crawlers, each responsible for collecting and cleaning related data from different sources.

The basic information crawler retrieves all GP names, postcodes and locations from the NHS GP page [17], where GP information is displayed in alphabetical order. The Google results crawler conducts searches using specific key information on the Google search engine, and collects the URLs of the corresponding GP overview pages. Then the GP overview crawler checks and processes the information on the overview pages obtained from the previous crawler, and extracts the specific GP website URLs, which are essential for the accessibility evaluation later. During each step, data cleaning is performed to refine and standardise the collected information.

3.3.1 Basic Information Crawler

This crawler is built using Scrapy, a web crawling and scraping framework in Python [20]. The basic structure of the crawler is shown in Figure 3.3.

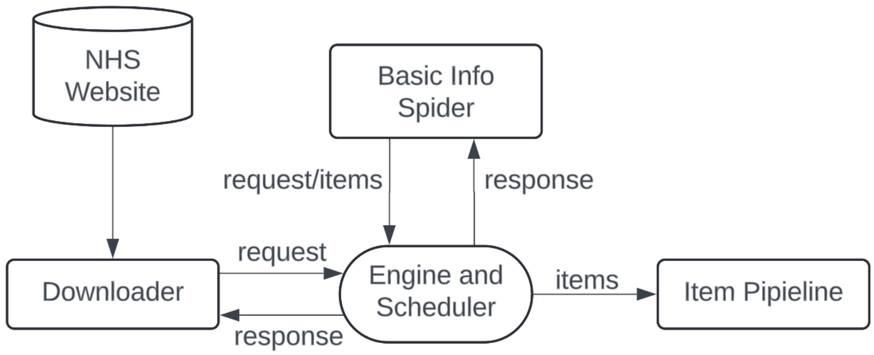

Figure 3.3: The overview of basic information crawler.

To initiate crawling, a starting URL is initialised and sent to the Scrapy Engine, which manages the data flow throughout the entire system. The Engine then dispatches the request to the Downloader, enabling it to retrieve web pages and subsequently return those pages to the Engine and then to the NHS Spider. Within the Spider, the relevant information is extracted and encapsulated into GP Item objects, which are structured and organised to facilitate further processing. These GP items are then transmitted to the Item Pipeline, which is responsible for data cleaning and persistence tasks. Throughout this process, the Scheduler receives and enqueues the requests in coordination with the Engine, ensuring efficient execution of the crawling tasks.

Data Collection

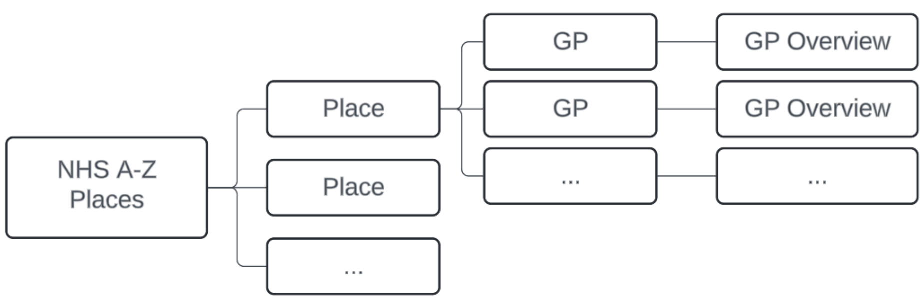

Figure 3.4: The overview of NHS website structure.

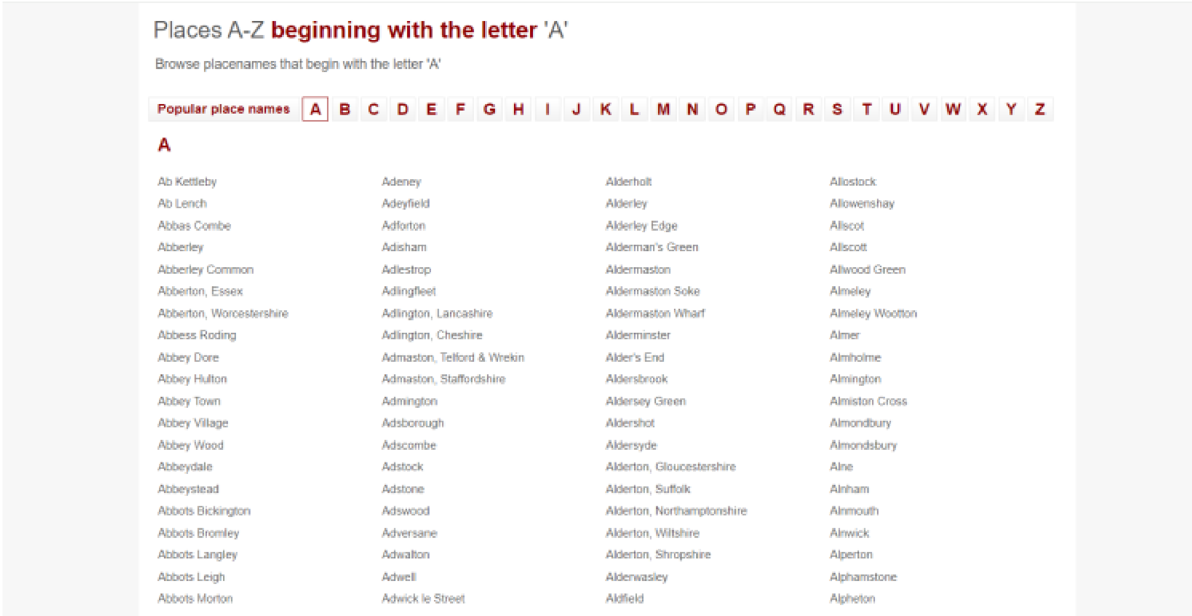

Figure 3.5: Screenshot of NHS A-Z places website displaying list of places. Source [Online] – available at: `https://www.nhs.uk/service-search/other-services/GP/Location/Places/A/4`.

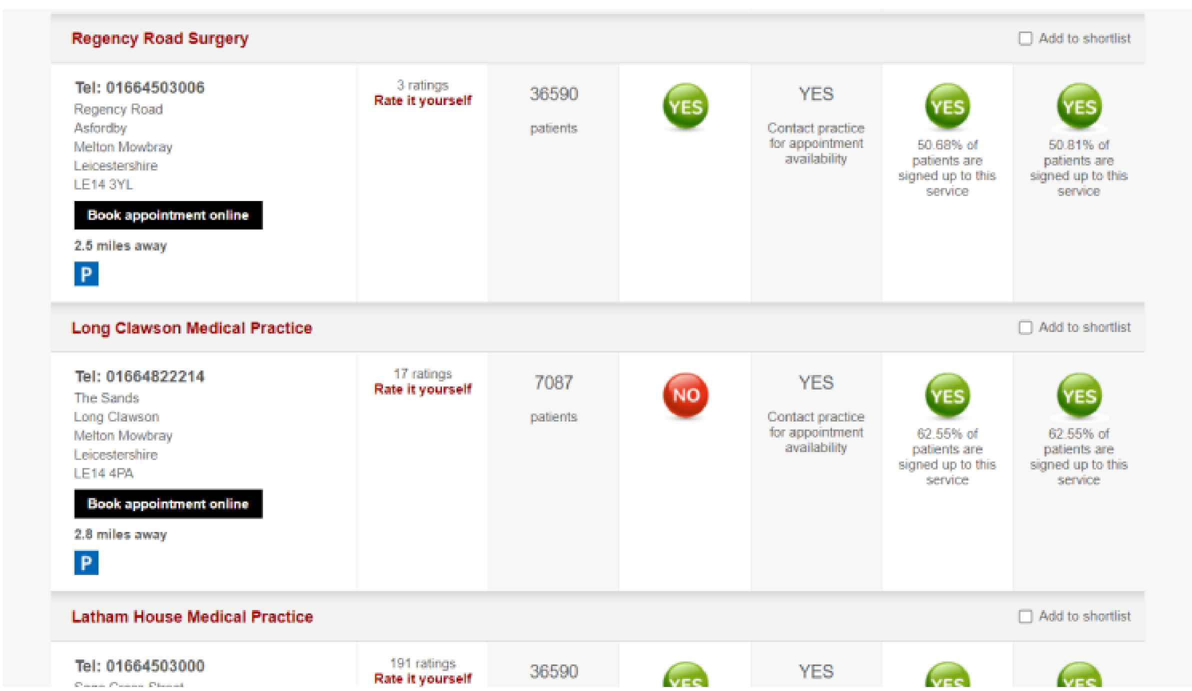

Figure 3.6: Screenshot of the GP list page in a place. Source [Online] – available at: `https://www.nhs.uk/service-search/other-services/GP/Ab-Kettleby/Results/4/-0.931/52.799/4/1?distance=25`.

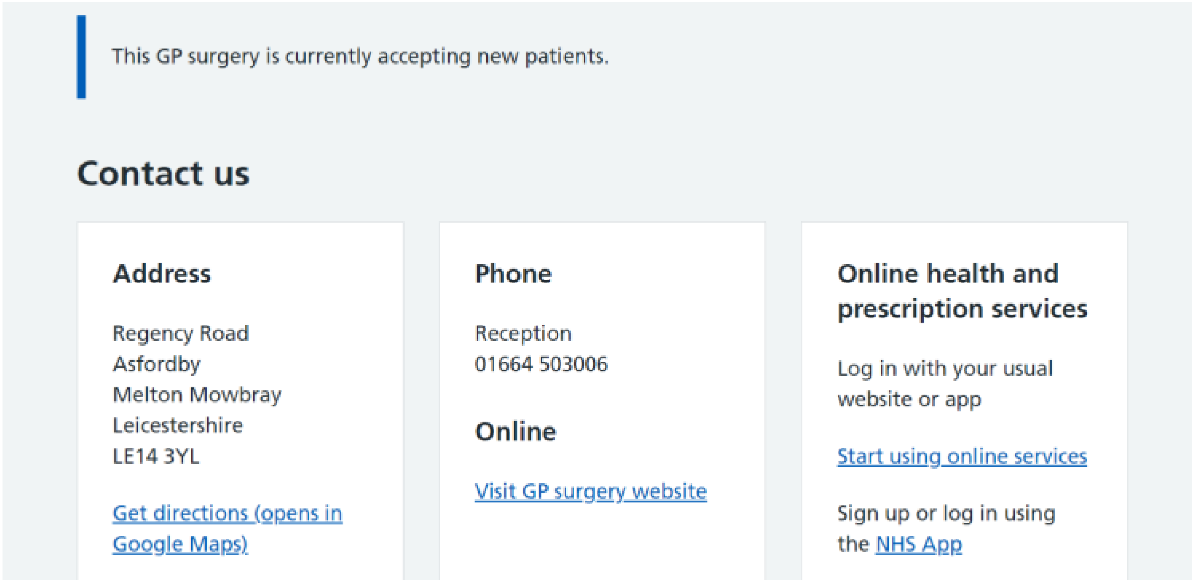

Figure 3.7: Screenshot of a random GP overview page. Source [Online] – available at: `https://www.nhs.uk/services/gp-surgery/regency-road-surgery/X35645?gsdServiceId=4`.

Figure 3.4 presents an overall structure of the NHS website related to GP searches. As shown in Figure 3.5, 3.6, and 3.7, the NHS GP places page encompasses a list of places sorted alphabetically. Each place contains a list of GPs accessible within a certain distance, although the GPs may not necessarily be physically located within that specific place. The original plan was to collect all the GP information at once, where the GP names, postcodes and locations could be found in the GP list page, while the website URLs could be extracted from the GP overview page.

However, issues emerged during the collection process as all links from the GP list page to each corresponding overview page were broken. As a result, retrieving GP URLs became infeasible. Hence this crawler was only able to gather basic GP information from the GP list page, including:

- **GP name:** The name of each GP. Within a single place, multiple GPs may share the same name.
- **GP postcode:** The postcode associated with each GP. In the same geographical area, GPs may share identical postcodes. The combination of GP name and postcode serves as a unique identifier for each GP.
- **GP location:** This refers to the location displayed in the last line of the address, typically denoting the county where the GP is situated. While this information was initially intended for use in map visualisation, it proved incompatible with the map data. Since the map uses local authority district names, which are different from the location names collected. Therefore, this information was not utilised in future analysis.

Data Cleaning

Given that the same GP may be accessible and listed in multiple places, the possibility of collecting duplicate GP items arises. To address this issue, a comparison process is implemented in the Item Pipeline for each GP item retrieved during the data collection phase. Specifically, the name and postcode of each GP item are compared against existing stored data. GPs with a unique combination of name and postcode are identified and stored, while duplicate items are omitted from further processing and storage.

3.3.2 Google Results Crawler

The crawler is constructed using Selenium, which enables browser automation [21]. It automates searches on the Google search engine using keywords based on the GP information stored in the previous step. Then it extracts the links to the overview pages for the corresponding GPs, which are essential for the subsequent GP URL collection task.

Data Collection

To ensure precise results from Google, the keyword is designed as "[GP postcode] [GP name] nhs gp overview", and is encoded to the URL as query parameters in the request. This tailored approach optimises the search query to retrieve the desired links of the overview pages.

Once the crawler receives the search results, it examines the links retrieved from the first page. Since links to GP overview pages commonly include the string "www.nhs.uk/services/gp-surgery", the crawler checks whether the link contains this string. If the string is detected, the link will be saved and checked in a later step to make sure the overview page and the desired GP match exactly.

Data Cleaning

The Python library pandas is used to clean the collected links. The cleaning process consists of the following steps:

1. **Delete missing links:** Each data entry that lacks a link to the overview page is identified and dropped from the dataset.

2. **Remove additional string endings:** Some URLs may include extra strings, such as "/leave-a-review", "/ratings-and-reviews" and "/how-to-register" added at the end. These extra components are undesirable for the overview page URLs, and thus are removed to obtain accurate overview page links.

3.3.3 GP Overview Crawler

The structure of the GP overview crawler is similar to that of the basic information crawler. This crawler mainly collects GP website URLs from the overview pages.

Data Collection and Cleaning

A comparison process is performed to confirm whether the overview page corresponds with the desired GP. The name and postcode details extracted from the overview page are compared to the database's existing records. If the combination of name and postcode matches the record, it confirms that the overview belongs to the specific GP.

Conversely, if no match is found between the extracted name-postcode combination and the existing records, it indicates that the overview does not belong to that specific GP. In such cases, the overview page is discarded without further processing.

After verifying the overview page, the GP website URL is extracted and stored. Additionally, more information is collected and stored into the database:

- **GP URL:** This refers to the official website link of each GP. If the overview page does not provide the URL, this will be marked as Null in the database.
- **State:** This field indicates the status of the GP website URL, and can be assigned to the values of -1, 0, 1 or Null, each corresponding to various scenarios:
  - 1: This indicates that the URL is working and can be evaluated at a later stage.
  - 0: This signifies that the URL is missing (not provided by the overview page).
  - -1: This denotes that although the URL is provided, it is not working and therefore inaccessible.
  - Null: This indicates that the overview page is not found, resulting in a missing website URL.

By including the additional field in the database, the system can effectively manage and track the status of GP website URLs, facilitating subsequent evaluation and analysis processes.

3.3.4 Data Collection Results

The data collection process requires approximately 200 hours to complete. Following data cleaning procedures, a total of 8613 unique GP records are collected. Among these records, 7103 URLs are successfully located and extracted from the overview pages. Subsequently, 6904 working URLs are identified and stored. The web accessibility of these URLs will be evaluated and analysed in the later steps.

3.4 Analysis Project

In the analysis project, web accessibility evaluation and analysis are conducted in the backend, and the results are visualised in the frontend. In the backend, the GP homepage URLs are retrieved from the database and sent to the AChecker API. The reports generated by the API are then stored in the database for further accessibility analysis. Moreover, a scheduler is implemented to automate the evaluation and analysis tasks on a regular basis, ensuring continuous monitoring of web accessibility. For the frontend, all the analysis results are visualised including an overall analysis page, a regional analysis page, a trend over time page, and a detailed result for individual GP. The whole project is then deployed and made accessible to all users.

3.4.1 Backend

AChecker API and Postcodes.io API

To facilitate subsequent web accessibility analysis, two APIs are incorporated into the system. The AChecker API accesses the web accessibility of the GP URLs, producing comprehensive reports. The Postcodes.io API provides additional geographical information, which is essential to the regional analysis and map visualisation process.

AChecker: This API is utilised to evaluate the web accessibility of GP websites. It accepts the GP URL as input, and generates an evaluation report in HTML format. Key information related to accessibility is extracted from the reports for further analysis, including:

- Number of errors, likely problems and potential problems.
- Number of errors violating A, AA and AAA level success criteria.
- Lists of IDs of errors, likely problems and potential problems.
- Lists of IDs of errors violating A, AA and AAA level success criteria.

Postcodes.io: This API is employed to retrieve the local authority district (LAD) names based on postcodes. This information is then used in the England map data [13], providing the boundaries and subnational divisions of England in map visualisation.

Database

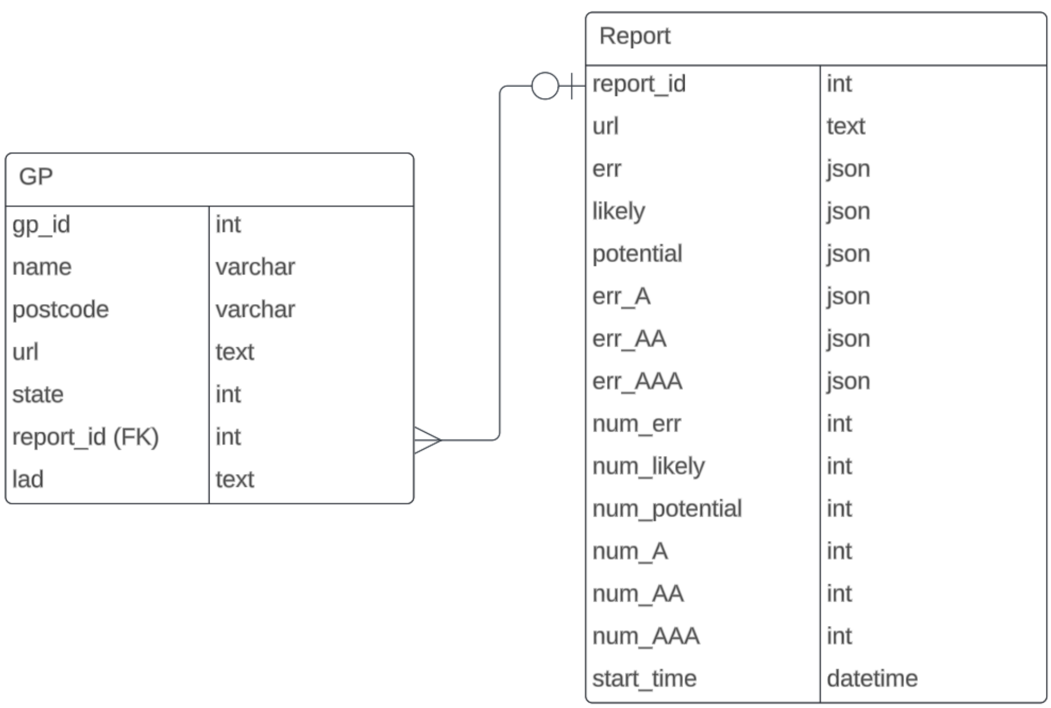

Figure 3.8: The database schema of the project.

In this analysis project, a MySQL database is integrated to organise the collected data, following the structure depicted in Figure 3.8. Each field within the GP table corresponds to specific attributes, detailed in Section 3.3.3.

Within the GP table, the report_id field serves as a foreign key, linking GP entries to their corresponding accessibility reports. In some scenarios, different GPs may have the same URL, and hence share the same accessibility report. However, in cases where the GP URL is not provided, the report_id field will be set to Null, indicating the absence of a linked accessibility report.

The Report table captures essential details extracted from the reports generated by the AChecker API. This includes information such as the number of issues and specific error IDs identified in the accessibility evaluation. Additionally, a starting time field is included to record the date when the URLs are evaluated and reports are produced.

Analysis

In conducting statistical analysis to gain insight into web accessibility across England's GP websites, the examination spans three dimensions: overall, regional, and individual GP.

Statistical analysis is conducted to get an insight into the web accessibility of England GP websites. The analysis is done in three dimensions: overall, regional and in individual GP.

For the overall analysis:

1. **Percentage of errors categorised by the guideline principles:** This aspect of the analysis involves calculating the percentage of errors that violate the four principles of web accessibility: Perceivable, Operable, Understandable and Robust.

2. **Identification of common errors:** The ten most frequent errors within each conformance level (A, AA, AAA) are identified across all GP websites. For each identified error, the number of occurrences and the violation in the corresponding criteria are also recorded.

For the district analysis:

1. **Average number of errors:** For each district in England, the analysis computes the average number of errors detected in the GP websites. This is achieved by summing the total number of errors and dividing it by the total number of GP websites in a district, as shown in Equation 3.1.

$$Average\ error = \frac{Number\ of\ errors}{Number\ of\ GPs} \tag{3.1}$$

   Similarly, the average number of likely and potential problems for each district is also calculated.

2. **Accessibility level calculation:** The accessibility level of each place is determined by calculating the percentage of GPs that satisfy each level of conformance. For example, the accessibility level for level A conformance is computed by dividing the number of GPs that satisfy level A conformance by the total number of GPs in a district, defined as Equation 3.2.

$$Accessibility\ level = \frac{Number\ of\ GPs\ that\ meet\ A\ conformance}{Number\ of\ GPs} \tag{3.2}$$

   Similar calculations are performed for AA and AAA conformance levels.

For the individual GP analysis:

1. **Number of each type of issue:** The analysis counts the number of issues including errors, likely problems and potential problems, for each individual GP.

2. **Error identification by conformance levels:** For each GP, the errors detected for the three conformance levels (A, AA, AAA) are gathered and sorted into separate tables. These tables provide detailed descriptions and causes for the specific problems.

Scheduler

To guarantee regular evaluation of web accessibility and timely updating of analysis results on the dashboard, Django APScheduler [12] is integrated into the system. This scheduler performs tasks at predetermined intervals within a Django project.

For each quarter of the year, the scheduler utilises the AChecker API to access the web accessibility and calls the analysis functions to process the latest reports returned by the API. The task is scheduled to be executed on the first day of months 1, 4, 7, and 10, at 00:00. This automated process monitors web accessibility across all GP websites and provides an up-to-date analysis of the current accessibility status.

3.4.2 Frontend

The dashboard is built using Apache ECharts [7], a powerful visualisation library, which offers a clear and interactive data visualisation with suitable animation. In addition, components and styling such as the navigation sidebar are implemented with Bootstrap [8]. These tools allow for the creation of a user-friendly dashboard and simple navigation.

Structure

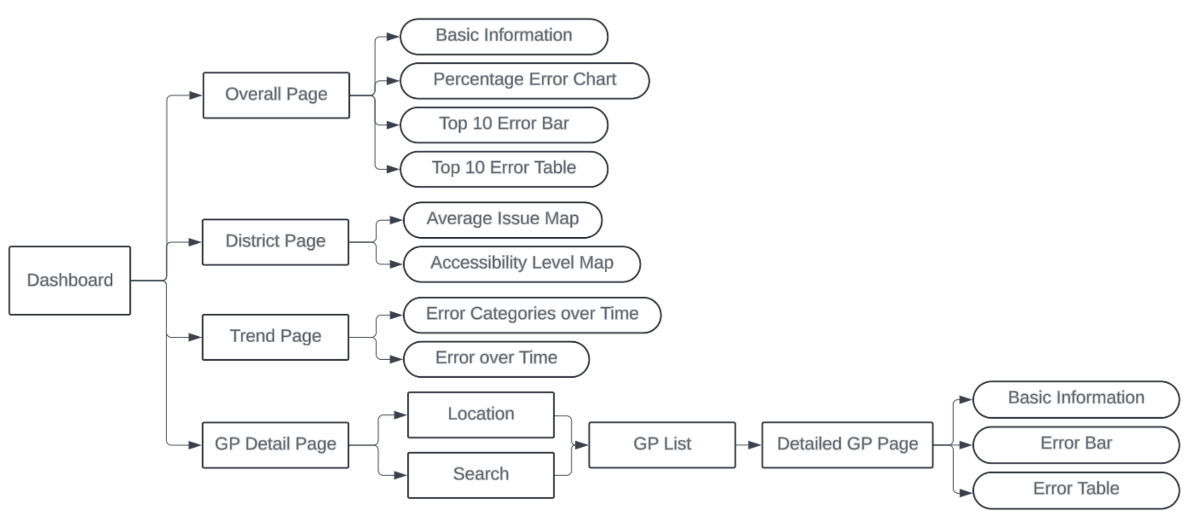

Figure 3.9: The architecture of the dashboard.

The dashboard is divided into four main sections as shown in Figure 3.9, each responsible for a specific aspect of analysis. The overall page displays an evaluation summary. It also contains a pie chart showing the percentage of errors under each category and a table of the most frequent errors. The district page presents accessibility data including the average number of errors and accessibility levels for each district on the England map. The trend page shows the number of different issues over time. In addition, the detailed GP page contains the evaluation results for each GP.

Interaction

The dashboard offers a range of interactive options to improve user experience and support in-depth analysis:

1. **Zooming and panning:** To focus on specific areas on the map, users can pan over the graph, zoom in and zoom out of the coordinate system.

2. **Filter values:** Users can specify a high or low threshold to search for locations using the filter bar that is provided beside the map.

3. **Restore graph size:** Users can choose to restore the graph to its initial size and reset the visualisation of a graph.

4. **Raw data review:** For each graph, users can view a list of raw data. This allows a transparent and detailed examination of the data points.

5. **Display or hide categories:** Users can select which categories to show or hide on charts. By toggling visibility for each category, users can easily focus their analysis on specific data points and compare between the chosen categories.

3.4.3 Deployment

The dashboard is deployed on the PythonAnywhere [6] platform, which provides a steady connection and minimal latency for users accessing the project.

3.5 Testing Methods

The testing process includes the automated tests of the Django project, and a manual usability evaluation conducted by the users.

3.5.1 Automated Tests

The automated testing consists of two main components: unit testing and functional testing. Django's unit test makes use of the Python standard library unittest [14]. In unit testing, the project systematically evaluates each function independently. This includes testing the analysis functions, API functions, as well as every model and URL within the project.

On the other hand, functional testing focuses on evaluating the project's behaviour and presentation from the viewpoint of end users. It utilises Selenium to simulate user interactions with the web pages in order to verify that particular elements and content operate correctly on each page.

3.5.2 Usability Evaluation

Twenty users are selected to participate in the usability evaluation by completing a questionnaire after interacting with the dashboard. Users are divided into two groups: one with a related technical background, and the other group consisting of those without domain knowledge. In the questionnaire, users are asked to complete some tasks and evaluate the complexity and difficulty of the tasks. Additionally, the System Usability Scale [10] is included in the questionnaire to provide a reliable assessment of usability. The survey is conducted using Microsoft Forms.

The tasks for users to complete are as follows:

1. For the overall analysis, which category has the most errors?
2. For the overall analysis, what are the IDs of the most common issues at each level (A, AA and AAA)?
3. For the district analysis, what is the average number of Errors in Liverpool?
4. For the district analysis, how many districts have more than 40 mean Errors (bar can be dragged to set threshold)?
5. For the trend analysis, what is the percentage of Operable errors in 2025 Q2?
6. For the trend analysis, which error category shows the most evident decreasing trend?
7. For the first GP (Ailsa Craig Medical Centre) listed in Manchester, how many potential problems does it have?
8. Find a GP with postcode “M14 6WP”, how many A level errors does it have?

The complete questionnaire can be found in Appendix A.

Chapter 4

Results and Evaluation

This chapter discusses the visualisation and the findings derived from the general and regional analyses detailed in Section 3.4.1. Then it provides the results of the evaluation process outlined in Section 3.5.2, which includes a questionnaire for usability evaluation.

4.1 Visualisation

Figure 4.1, 4.2, and 4.3 depict the visualisation for the analysis in accordance with the components listed in Section 3.4.2, under the overall page, districts page, and detailed GP report page.

Overall Page (Figure 4.1): This web page presents an overview of web accessibility for all GP websites, including the percentage of errors categorised by guideline principles and the identification of common errors.

District Page (Figure 4.2): This page provides insights into the accessibility performance of GP websites across different districts. This includes the average number of errors, likely problems, and potential problems for each district, as well as the accessibility level of each district based on conformance levels.

Detailed GP Report Page (Figure 4.3): This page offers detailed visualisation related to individual GP reports. Users can delve into specific GP and examine different types of issues identified.

Trend Page (Figure 4.4): In this page, the trend of errors categorised by the four guideline principles for each year quarter is visualised using a line plot. Additionally, an accompanying pie chart illustrates the percentage of errors violating each principle for each quarter. The bar chart displays the number of various types of issues (including errors, likely problems, potential problems and errors violating three conformance levels) over each quarter, with a polyline indicating the trend for each type of issue over time.

The project is designed to automatically update accessibility results on a quarterly basis. However, due to time constraints, data collection for subsequent quarters is not completed. As a temporary measure, dummy data is utilised to simulate the results for future quarters.

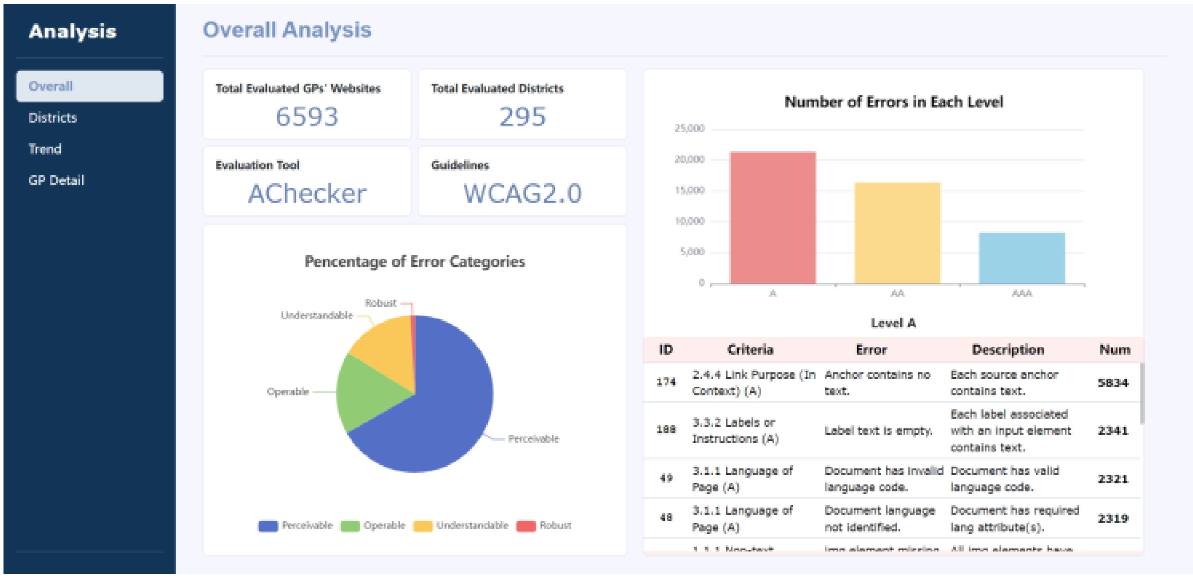

Figure 4.1: Screenshot of overall analysis page.

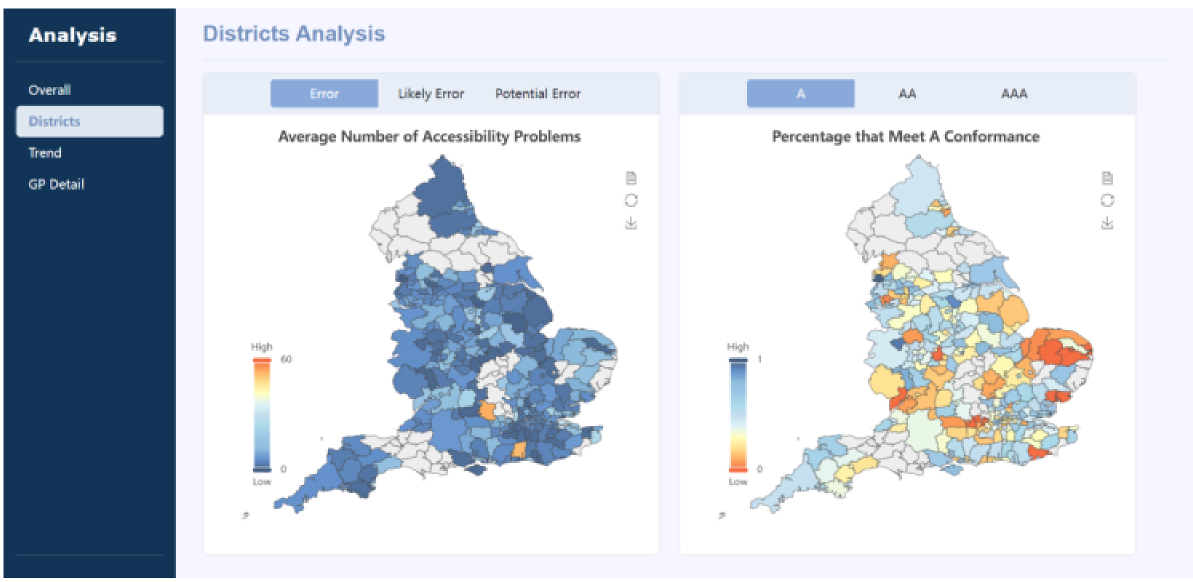

Figure 4.2: Screenshot of district analysis page.

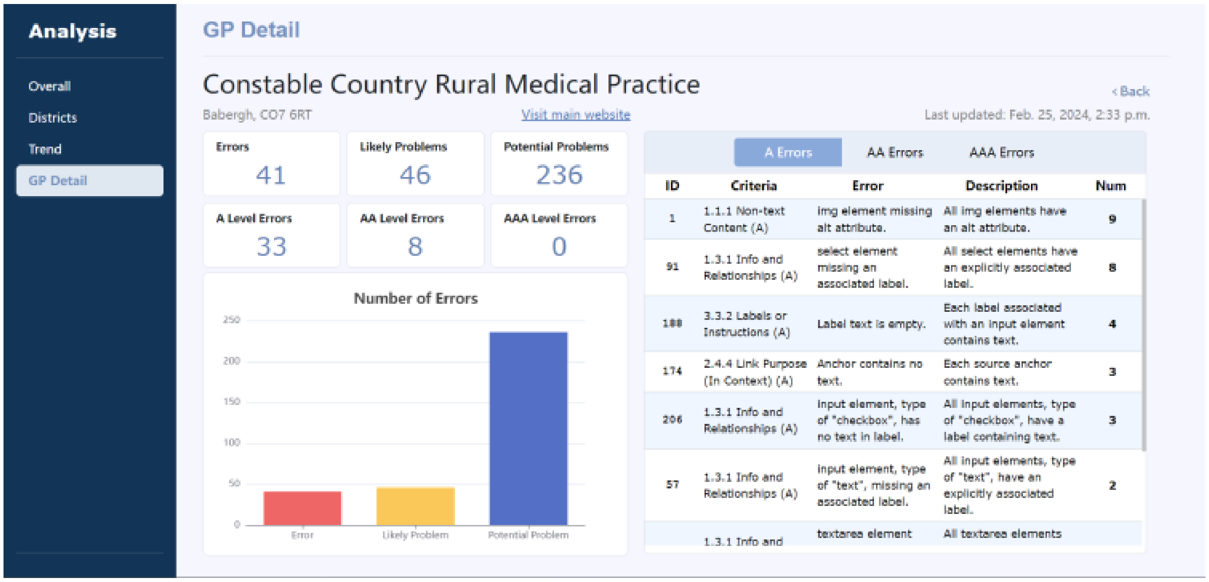

Figure 4.3: Screenshot of the detailed report page.

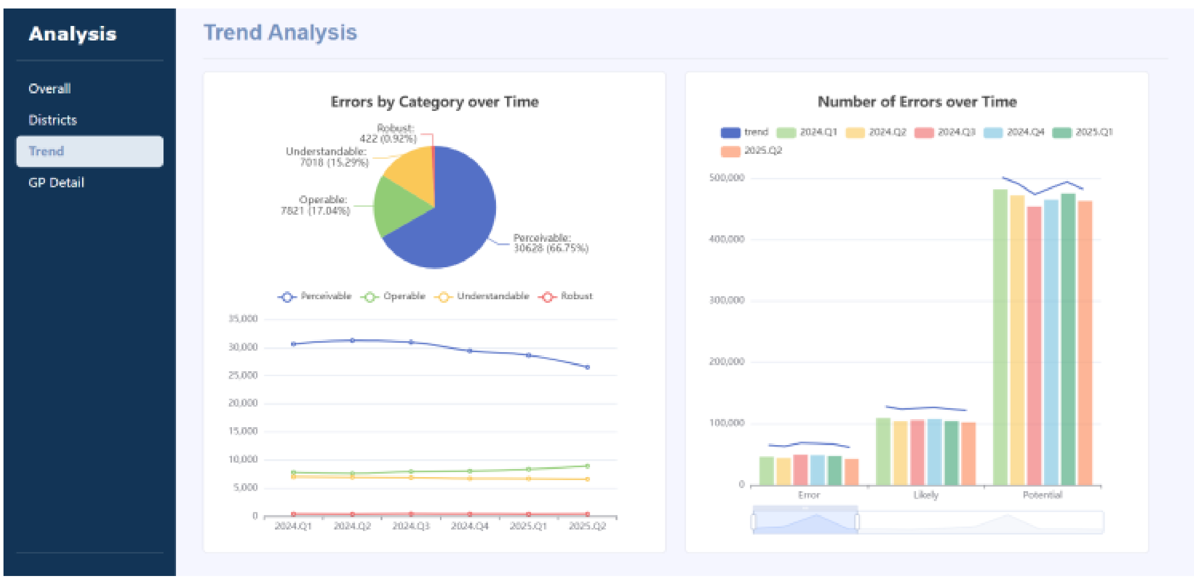

Figure 4.4: Screenshot of trend analysis page.

Figure 4.5: Screenshot of search result with keyword “manchester”.

Figure 4.5 shows an example of the search results obtained using the keyword “manchester”. Users have the flexibility to input the keyword based on GP name, postcode, or location to retrieve desired GPs. The search functionality is case-insensitive. Furthermore, the results are presented with the highlighted keyword, allowing easy identification and efficient navigation.

By organising the dashboard into these sections and providing corresponding visualisation, users can easily navigate through the analysis findings and gain comprehensive insights into the accessibility status of GP websites from different dimensions.

4.2 Accessibility Analysis

In the analysis, a total of 6,593 GP Websites are evaluated, comprising 6,006 unique URLs, across 295 local authority districts based on WCAG 2.0 guidelines. As shown in Figure 4.6, the evaluation detected a total of 45,889 errors, with the majority under the category of Perceivable (30,628 errors, 66.75%), followed by Operable (7,821 errors, 17.04%), Understandable (7,018 errors, 15.29%), and Robust (422 errors, 0.92%). Additionally, 108,843 likely problems and 108,843 potential problems are identified. Among the evaluated websites, 2,220 URLs have no detected errors, with 1,097 of them also reporting zero likely or potential problems. However, one of the GP websites with the highest number of known problems is the Berinsfield Health Centre site, recording a total of 765 errors. These issues are distributed across different conformance levels, with 623 errors violating A level criteria, 24 violating AA level criteria, and 117 violating AAA level criteria. The most commonly violated criterion for this website is “1.3.1 Info and Relationships”, which is attributed to “`input` element that contains a type attribute value of ‘radio’ does not have text in its associated label”.

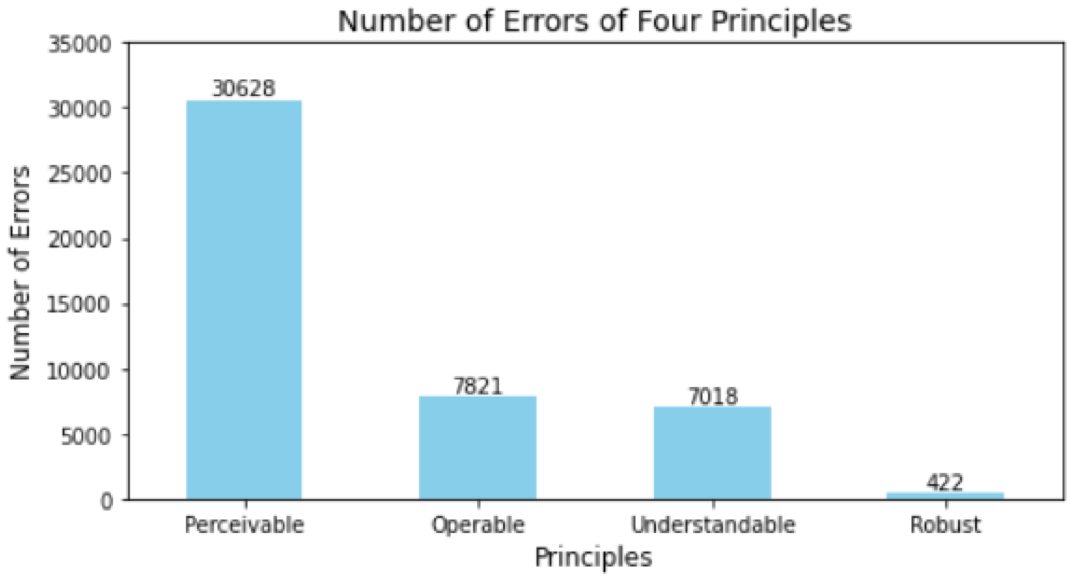

Figure 4.6: The bar chart for number of errors categorised by principles of Perceivable, Operable, Understandable and Robust.

Among all the known problems, the number of errors for A, AA and AAA conformance levels are 21,301 for 16,313, 8,245, respectively. The most common errors identified at each level are listed in Table 4.1, 4.2 and 4.3. For level A conformance, the most frequent errors include:

1. "Anchor (`a`) contains no text": Websites with this error typically lack text in the title attribute of the anchor element. Alternatively, for an image in the anchor element, the alt attribute is missing.

2. "Label text is empty": Some websites lack label text associated with an input element.

3. "Document contains invalid language code": This may occur because the website contains an invalid lang attribute.

These errors violate criteria related to "2.4.4 Link Purpose (In Context)", "3.3.2 Labels or Instructions" and "3.1.1 Language of Page". This indicates that the errors at level A mainly violate Operable and Understandable aspects of web accessibility.

For level AA conformance, the most common errors include:

1. "Italic (`i`) is used" and "Bold (`b`) is used": These errors are typically generated when websites use the `i` and `b` elements for italic and bold text respectively. Screen readers may not interpret these elements correctly as emphasis elements like `em` and `strong`. It is recommended to replace `i` and `b` with `em` and `strong` elements. These issues are widespread across websites, with a count of over 14,000.

2. "Header nesting is incorrect": This error occurs due to incorrect header structure, where the following header should be equal to, one level greater or any level less than the current header level.

The majority of these errors violate the criteria "1.4.4 Resize Text" and "2.4.6 Heading and Labels", which are related to Perceivable and Operable principles.

For level AAA conformance, the most common errors are closely related to contrast issues, violating criterion “1.4.6 Contrast (Enhanced)”. These errors are typically caused by insufficient contrast between text colour and background colour, or between the colour of links in different states (including visited, selected and active) and the background colour.

Table 4.1: A list of the most common accessibility errors at level A.

ID	Criteria	Error	Description	Num
174	2.4.4 Link Purpose (In Context) (A)	Anchor contains no text.	Each source anchor contains text.	5834
188	3.3.2 Labels or Instructions (A)	Label text is empty.	Each label associated with an `input` element contains text.	2341
49	3.1.1 Language of Page (A)	Document has invalid language code.	Document has valid language code.	2321
48	3.1.1 Language of Page (A)	Document language not identified.	Document has required `lang` attribute(s).	2319
1	1.1.1 Non-text Content (A)	`img` element missing `alt` attribute.	All `img` elements have an `alt` attribute.	2108
7	1.1.1 Non-text Content (A)	Image used as anchor is missing valid `alt` text.	`alt` text for all `img` elements used as source anchors is not empty when there is no other text in the anchor.	1585
204	1.3.1 Info and Relationships (A)	`input` element, type of “radio”, has no text in the label.	All `input` elements, type of “radio”, have a label containing text.	1032
57	1.3.1 Info and Relationships (A)	`input` element, type of “text”, missing an associated label.	All `input` elements, type of “text”, have an explicitly associated label.	750
121	1.3.1 Info and Relationships (A)	`input` element, type of “radio”, missing an associated label.	All `input` elements, type of “radio”, have an explicitly associated label.	728
213	1.3.1 Info and Relationships (A)	`input` element, type of “text”, has no text in the label.	All `input` elements, type of “text”, have a label containing text.	680

Table 4.2: A list of the most common accessibility errors at level AA.

ID	Criteria	Error	Description	Num
117	1.4.4 Resize text (AA)	`i` (italic) element used.	`i` (italic) element is not used.	13332
116	1.4.4 Resize text (AA)	`b` (bold) element used.	`b` (bold) element is not used.	1053
38	2.4.6 Headings and Labels (AA)	Header nesting - header following `h2` is incorrect.	The header following an `h2` is `h1`, `h2` or `h3`.	838
37	2.4.6 Headings and Labels (AA)	Header nesting - header following `h1` is incorrect.	The header following an `h1` is `h1` or `h2`.	759
177	1.4.4 Resize text (AA)	`font` used.	`font` must not be used.	218
39	2.4.6 Headings and Labels (AA)	Header nesting - header following `h3` is incorrect.	The header following an `h3` is `h1`, `h2`, `h3` or `h4`.	127
40	2.4.6 Headings and Labels (AA)	Header nesting - header following `h4` is incorrect.	The header following an `h4` is `h1`, `h2`, `h3`, `h4` or `h5`.	16

Table 4.3: A list of the most common accessibility errors at level AAA.

ID	Criteria	Error	Description	Num
306	1.4.6 Contrast (Enhanced) (AAA)	The contrast between the colour of text and its background is not sufficient to meet WCAG 2.0 Level AAA.	Text colour must provide high contrast with its background colour	7535
309	1.4.6 Contrast (Enhanced) (AAA)	The contrast between the colour of selected link text and its background is not sufficient to meet WCAG 2.0 Level AAA.	Selected link text colour must provide high contrast with its background colour.	603
308	1.4.6 Contrast (Enhanced) (AAA)	The contrast between the colour of active link text and its background is not sufficient to meet WCAG 2.0 Level AAA.	Active link text colour must provide high contrast with its background colour.	48
307	1.4.6 Contrast (Enhanced) (AAA)	The contrast between the colour of visited link text and its background is not sufficient to meet WCAG 2.0 Level AAA.	Visited link text colour must provide high contrast with its background colour.	29
310	1.4.6 Contrast (Enhanced) (AAA)	Link text colour must provide high contrast with its background colour.	The contrast between the colour of link text and its background is not sufficient to meet WCAG 2.0 Level AAA.	25

In the analysis, out of the total number of evaluated websites, 2,353 meet level A conformance, 2,233 meet level AA conformance, and 2,220 meet level AAA conformance. This indicates that a significant majority of GP websites (60.8%) fail to satisfy even the minimum level of conformance. Additionally, the percentage of GP websites that meet each level of criteria is calculated for each district, as detailed in Section 3.4.1. Figure 4.7 shows the result for level A conformance, where the highest percentage is 1, signifying that every GP website achieves A level conformance.

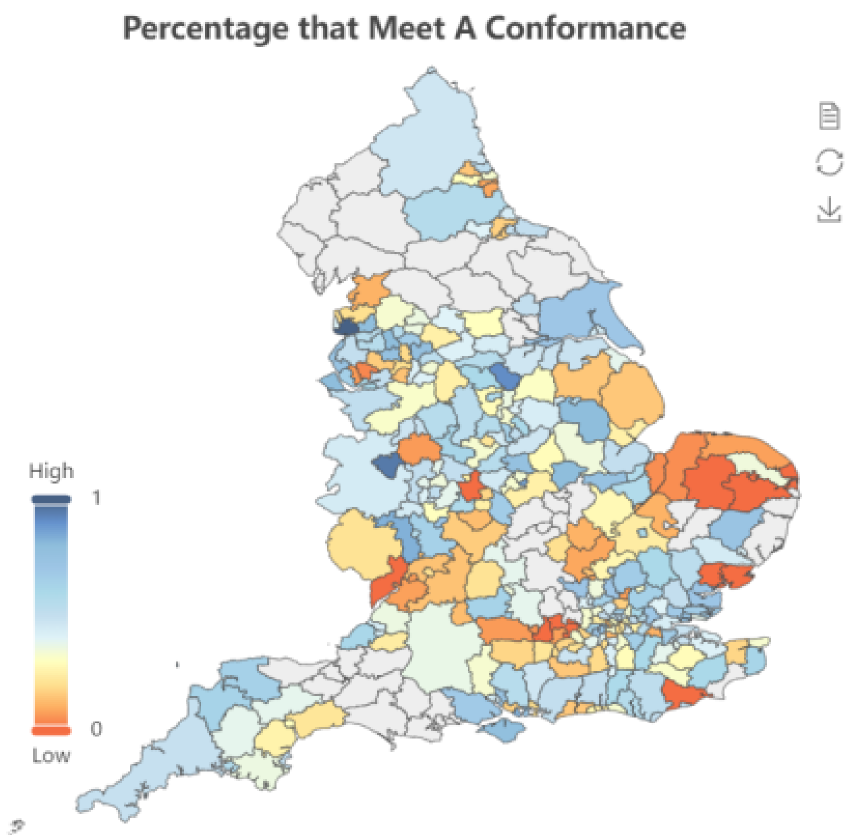

Figure 4.7: The percentage of websites that satisfy level A conformance for each region.

Figure 4.8 illustrates the top ten districts with the highest percentage accessibility for level A conformance. It is notable that Fylde, Telford and Wrekin, and Rotherham stand out with the highest percentage accessibility (over 90%), indicating that a majority of GPs in these areas satisfy all level A criteria. In most of the ten regions, the number of GP websites is less than 20. Additionally, these websites tend to adopt similar website designs provided by organisations such as MSW Digital Practice and iatro. These organisations offer usable and accessible GP website designs approved by the NHS, which may assist GPs in building websites compliant with AA and even AAA levels of accessibility.

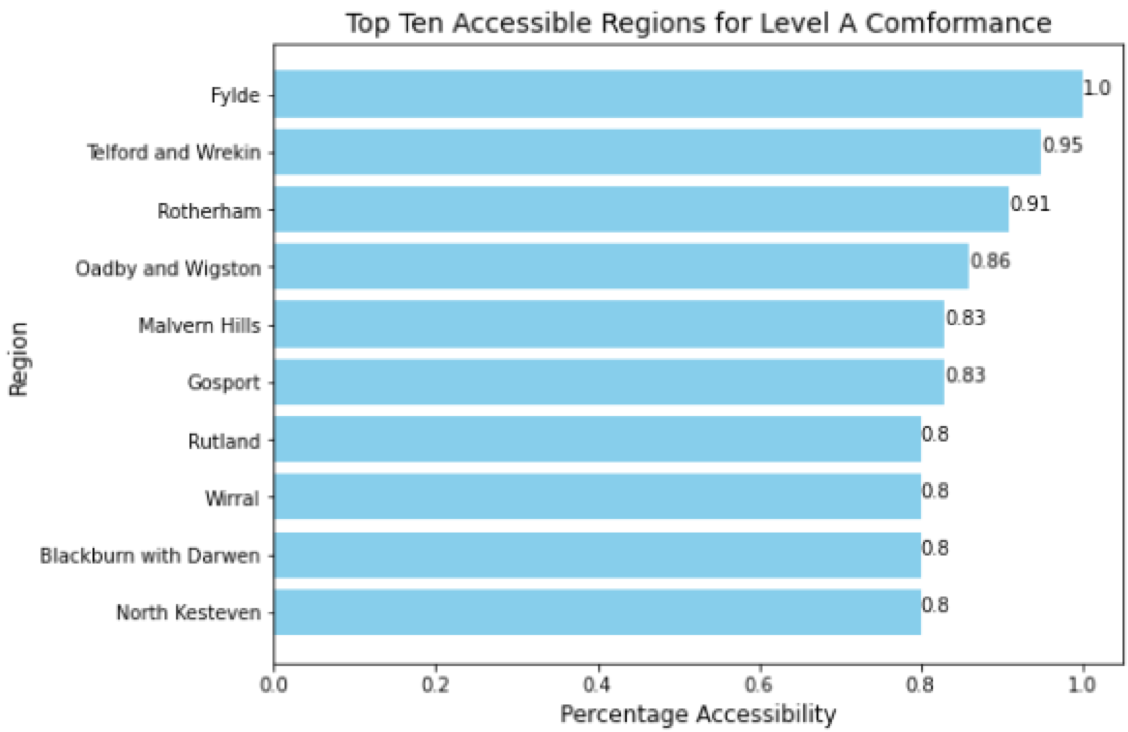

Figure 4.8: The top ten accessible regions based on level A conformance.

However, as shown in Table 4.4, twelve districts, including Forest of Dean, Great Yarmouth, and Breckland, show 0 percent accessibility. This indicates that none of the GP websites in these areas meet A level conformance. These websites have common errors like "anchor contains no text", even though they may have fewer accessibility issues overall. Since these websites generally utilise a similar design, every website reports a similar set of errors.

Table 4.4: Twelve regions with zero percent accessibility based on level A conformance.

District	Percentage Accessibility
Forest of Dean	0
Great Yarmouth	0
Breckland	0
South Norfolk	0
Tendring	0
Rother	0
North Warwickshire	0
Slough	0
Norwich	0
Wokingham	0
Colchester	0
Windsor and Maidenhead	0

4.3 Usability Evaluation

In this section, the outcomes of the usability evaluation process explained in Section 3.5.2 are presented. Twenty users are chosen to complete the questionnaire. Within this group, ten of them have relevant knowledge (with a background in computer science or related fields), while the remaining ten are from non-technical backgrounds. The questionnaire is divided into two distinct sections. In the first part, users are tasked with completing specific actions on the website, while the subsequent includes SUS questions designed to reflect users' subjective impressions of the system's usability and effectiveness.

4.3.1 Questions

As mentioned in the previous section, the first task comprised eight questions, each showing high accuracy rates across all twenty users, as presented in Table 4.5:

Table 4.5: Accuracy for the eight questions.

Questions	Accuracy
For the overall analysis, which category has the most errors?	100%
For the overall analysis, what are the IDs of the most common issues at each level (A, AA and AAA)?	90%
For the district analysis, what is the average number of Errors in Liverpool?	100%
For the district analysis, how many districts have more than 40 mean Errors (bar can be dragged to set threshold)?	100%
For the trend analysis, what is the percentage of Operable errors in 2025 Q2?	100%
For the trend analysis, which error category shows the most evident decreasing trend?	100%
For the first GP (Ailsa Craig Medical Centre) listed in Manchester, how many potential problems does it have?	100%
Find a GP with postcode "M14 6WP", how many A level errors does it have?	100%

The answers to the first two questions were accessible on the overall analysis page, which were clearly displayed in the pie chart and the error list. Incorrect responses may be selected when users chose the three most common errors in level A table, rather than examining the individual tables for levels A, AA, and AAA separately. However, the majority of users answered correctly, indicating that the visualisation on the overall page was sufficiently clear.

For questions related to the district analysis page, users were tasked with locating Liverpool on the map to obtain the corresponding number of errors. Although users who were unfamiliar with UK geography may require additional time for research, all participants provided correct answers to this question. Similarly, the second question on the district page required users to utilise the filter bar beside the map. Impressively, all users successfully manipulated the bar to obtain the filtered data they desired, demonstrating the usability and clarity of the map functionality provided by the district page.

The two questions related to the trend analysis page could be found easily according to the pie chart and line plot provided, resulting in the high accuracy as expected. The last two questions were related to the GP detailed page. Users could navigate to a specific GP detail page by either checking the location on the location page and locating the desired GP in the GP list, or directly searching the name, location, or postcode of the GP using the search bar. The notable accuracy of these questions implies that users from diverse backgrounds can easily acquire detailed accessibility results for specific GP websites.

In summary, the website offers highly usable functionality with clear and understandable visualisation for users from various backgrounds and domains.

4.3.2 SUS

The second part of the questionnaire is based on the System Usability Scale, which measures effectiveness, efficiency and satisfaction from users in a time-efficient manner. SUS consists of ten statements concerning the user experience with the system. Users indicate their level of agreement with each statement on a scale from 1 to 5, ranging from strong disagreement to strong agreement. The average results for each question completed by the twenty participants can be summarised and listed in Table 4.6, Also, Figure 4.9 displays the score distribution for each question.

Table 4.6: Average scale for the SUS questions.

Question	Average Scale
1	3.65
2	1.70
3	4.45
4	1.70
5	4.45
6	1.60
7	4.35
8	1.30
9	4.15
10	1.85

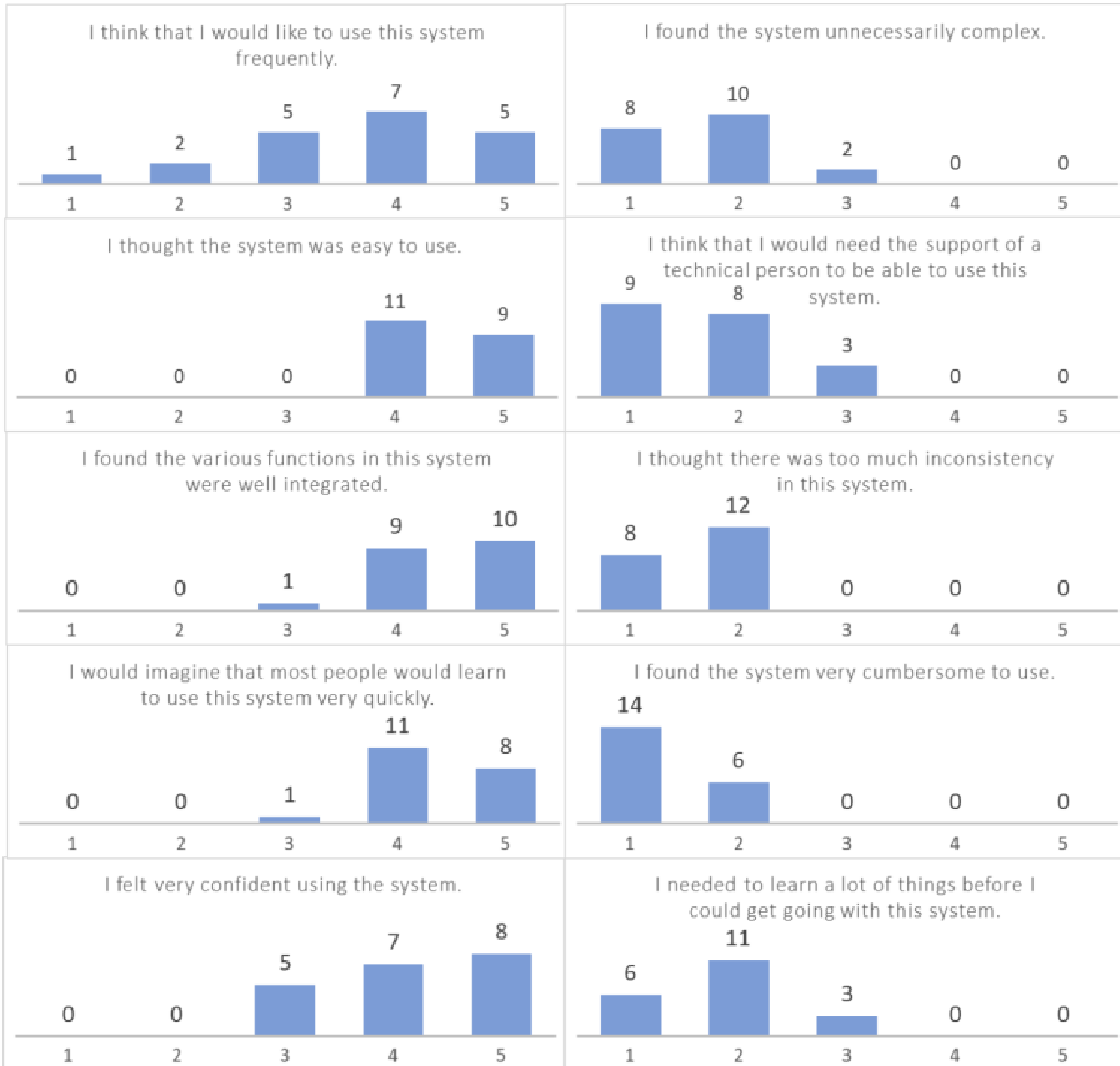

Figure 4.9: The score distribution of each question.

The final score is calculated according to the scoring method provided by John Brooke [10]. For questions 1, 3, 5, 7 and 9, the score is derived by subtracting 1 from the chosen scale. For questions 2, 4, 6, 8, and 10, the score equals 5 minus the chosen scale. The resulting score is then scaled to range from 0 to 100 by multiplying by 2.5. Therefore, using the average scale, the scoring equation can be written as below, in Equation 4.1:

$$score = 2.5 \times \left(\sum_{i \in odd\, index} scale(i) - 5 + 25 - \sum_{i \in even\, index} scale(i) \right) \tag{4.1}$$

Hence the final score is 82.25. To quantitatively evaluate the result, the Curved Grading Scale for the SUS [16] is utilised. The complete grading scale table is presented in Table 4.7. As observed from the table, the final score falls within the range of 80.8 to 84.0, corresponding to Grade A, with a percentile range exceeding 90%. This indicates the website provides an above-average user experience.

Table 4.7: Grading Scale for SUS.

Grade	SUS	Percentile range
A+	84.1 - 100	96 - 100
A	80.8 - 84.0	90 - 95
A-	78.9 - 80.7	85 - 89
B+	77.2 - 78.8	80 - 84
B	74.1 - 77.1	70 - 79
B-	72.6 - 74.0	65 - 69
C+	71.1 - 72.5	60 - 64
C	65.0 - 71.0	41 - 59
C-	62.7 - 64.9	35 - 40
D	51.7 - 62.6	15 - 34
F	0 - 51.6	0 - 14

Chapter 5

Conclusion

This chapter summarises the objectives achieved by the project. It also explores its limitations and potential future improvements. Finally, the chapter concludes with a self-reflection, discussing difficulties overcome and lessons learnt throughout the development of this project.

5.1 Achievements

The project successfully evaluates the web accessibility of all NHS GP websites in England and develops a dashboard to provide updated accessibility results every quarter. The key accomplishments include:

1. **Development of crawlers:** Three crawlers were built to efficiently collect relevant GP information. GP names, postcodes, and URLs were cleaned and stored in the MySQL database.
2. **Integration with AChecker:** AChecker was embedded into the system to automatically generate detailed reports for each GP website.
3. **Statistical analysis:** Analysis was performed on the accessibility results based on WCAG 2.0 to extract meaningful insights and trends.
4. **Dashboard creation:** A user-friendly dashboard was built to visualise the accessibility results from general, regional and individual GP perspectives.
5. **Quarterly evaluation:** The evaluation and analysis tasks were performed quarterly to update and monitor the accessibility level across all GP websites.

The practical usage of this project may benefit various users:

1. **Healthcare providers:** The project effectively monitors the accessibility of their GP websites and obtains individual accessibility reports generated for each GP. This allows healthcare providers to identify areas of improvement and take proactive steps to improve web accessibility. Furthermore, it becomes possible to compare with other practices within the same district or across England, which motivates healthcare organisations to prioritise web accessibility improvements.
2. **Individuals:** Older people and individuals from diverse backgrounds and abilities can benefit from improved access to online healthcare resources with a better user experience.

3. **Society:** This project may also directly or indirectly contribute to inclusivity and equity in healthcare access for a broader community, especially for people with disabilities, and underserved populations.2 Limitations and Improvements

5.2 Limitations and Improvements

Indeed, while this project achieves all the objectives, there are still limitations and areas for improvement.

5.2.1 Data Collection

In this project, the data collection process involves a multi-step pipeline to gather all relevant GP information. As mentioned in Section 3.3, the process encounters challenges including temporary link failures. Therefore, three separate crawlers are developed to ensure complete data retrieval. This approach significantly increases the complexity and difficulty of the data collection task. Additionally, when searching for GP information from the search engine, there is a risk of getting incorrect and undesirable information, which requires additional validation steps.

Improvements to the data collection process could accelerate operations and enhance data correctness. Once the links provided by NHS websites are restored, the data collection process can be largely simplified. With the restored links, it becomes feasible to collect more accurate information with reduced effort and time.

5.2.2 Analysis Scope

The project mainly collects basic information such as names, postcodes, locations, and URLs for each GP. However, other relative data from NHS websites could also be gathered. For example, information on the number of registered patients and website ratings could provide valuable insights into patient satisfaction with web accessibility. In addition, considering factors such as economics, population and regional size would allow for a deeper understanding of the relationship between web accessibility and various socioeconomic factors. This analysis could uncover patterns in accessibility levels across different regions, beneficial for addressing web accessibility challenges in different areas.

Moreover, while the current evaluation only assesses the homepage of each GP website, further research could also evaluate other web pages from the same site. This would provide users with a more thorough assessment of web accessibility across several GP website components, improving the usefulness of the analysis.

5.2.3 Accessibility Correctness

In this project, the embedded tool AChecker mainly analyses the HTML elements of the GP websites to assess web accessibility. However, web accessibility is not solely determined by HTML. It also depends on a website's CSS and JavaScript, which provide additional presentation and interactivity. Hence including CSS and JavaScript in the accessibility evaluation process may produce a more reliable result.

Furthermore, some websites use widgets provided by organisations like UserWay to provide additional accessibility. These widgets allow users to customise the website's contrast, text spacing, and cursors. The accessibility provided by these widgets may impact the overall accessibility of the website, and should be thoroughly analysed. Therefore taking CSS, JavaScript and accessibility widgets into consideration will result in a more accurate evaluation of web accessibility across GP websites.

5.2.4 Accessibility Evaluation Method

In this project, AChecker is the only evaluation tool adopted in the system. However, relying on a single tool may not be sufficient for achieving high-quality evaluation results. A single tool may have limited capability to evaluate every criterion extensively, thus may produce false positive or false negative results.

To address this limitation, future efforts may focus on comparing and integrating multiple accessibility evaluation tools into the system. The agreement and disagreement among these tools can be recorded and analysed. In addition, the scoring metrics can aid in quantifying the degree of agreement among the tools. In addition to the automated evaluation tools, human judgment is also essential for improving the accuracy of the analysis results. Automated tools, including AChecker, may have various levels of accuracy and coverage. For instance, AChecker also reports likely problems and potential problems, which require manual checks to further decide whether the error is present or not. Moreover, a confusion matrix of the evaluation reports may help assess the quality of the generated results. Through the combination of automated tools and human decision, the system may accomplish a more exhaustive and unbiased study of the web accessibility evaluation.

5.2.5 Website Improvement

User feedback can be used to improve the dashboard and reflect on their general satisfaction and challenges. Although the usability evaluation carried out for this project offers a summary of the website's usability, further extensive testing with a wider range of user groups could validate the website's effectiveness and usability. A more user-friendly dashboard could be designed by considering more user testing and feedback.

5.3 Reflection

During the development of this project, several challenges emerged:

1. **Lack of valid URLs:** As mentioned in Section 3.3, the NHS website did not provide a list of working URLs for the GP websites. To bypass this barrier, further work was done to search and collect the relevant GP website URLs from the Google search engine.

2. **Duplicate data collection:** In the initial collection phase, a large amount of duplicate GP information was collected incorrectly. This occurred because a GP can be accessible from different locations within a certain distance and therefore may appear in multiple places' lists. However, this aspect was not initially considered during the collection process, leading to approximately 8 million duplicate rows of data. Subsequently, the redundant data had to be discarded. To address this issue, a filtering mechanism based on GP

name and postcode was implemented to identify and remove duplicate entries (described in Section 3.3.1). In the end, around 7,000 unique URLs were collected successfully.

3. **Dashboard development:** Due to the lack of previous experience in building a visualisation dashboard, related knowledge had to be learnt from scratch. For the frontend, the JavaScript visualisation libraries were evaluated and compared. Echarts was chosen because it provides a more clear and interactive charting and visualisation style. On the backend, the testing methods for the Django project were learnt from the beginning. Unit tests and functional tests were embedded to programmatically examine the system's functionality and ensure it worked as expected.

In conclusion, while this project represents an advancement in the web accessibility evaluation of the NHS GP websites, it is clear that further development and adaptation are necessary. By continuing to explore potential improvements, it aims to contribute to an inclusive digital healthcare ecosystem for everyone.

References

[1] J. Abascal, M. Arrue, and X. Valencia. Tools for web accessibility evaluation. *Web accessibility: a foundation for research*, pages 479–503, 2019.

[2] S. G. Abduganiev. Towards automated web accessibility evaluation: a comparative study. *International Journal of Information Technology and Computer Science*, 9(9):18–44, 2017.

[3] P. Acosta-Vargas, S. Luján-Mora, T. Acosta, and L. Salvador-Ullauri. Toward a combined method for evaluation of web accessibility. In *Proceedings of the international conference on information technology & systems (ICITS 2018)*, pages 602–613. Springer, 2018.

[4] P. Acosta-Vargas, S. Luján-Mora, and L. Salvador-Ullauri. Evaluation of the web accessibility of higher-education websites. In *2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–6. IEEE, 2016.

[5] Y. Akgül and K. Vatansever. Web accessibility evaluation of government websites for people with disabilities in turkey. *Journal of advanced management science*, 4(3), 2016.

[6] Anaconda. Pythonanywhere. `https://www.pythonanywhere.com/`, 2024.

[7] Apache Software Foundation. Apache echarts. `https://echarts.apache.org/en/index.html`, 2024.

[8] Bootstrap team. Bootstrap. `https://getbootstrap.com/`, 2024.

[9] G. Brajnik. Validity and reliability of web accessibility guidelines. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pages 131–138, 2009.

[10] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[11] Cantan Group. Web accessibility checker. `https://websiteaccessibilitychecker.com/documentation/index.php?p=checker/index.php`, 2023.

[12] J. Cass. Django apscheduler. `https://pypi.org/project/django-apscheduler/`, 2022.

[13] M. Chorley. Uk-geojson. `https://github.com/martinjc/UK-GeoJSON`, 2016.

[14] Django Software Foundation. Writing and running tests. https://docs.djangoproject.com/en/5.0/topics/testing/overview/, 2024.

[15] S. Harper and A. Q. Chen. Web accessibility guidelines: A lesson from the evolving web. *World Wide Web*, 15:61–88, 2012.

[16] J. R. Lewis and J. Sauro. Item benchmarks for the system usability scale. *Journal of Usability Studies*, 13(3), 2018.

[17] National Health Service. Places a-z. https://www.nhs.uk/service-search/other-services/GP/Location/Places/A/4, 2024.

[18] G. Ramiro, M. José, P. Jorge, P. Carlos, R. Tânia, and F. Branco. Accessweb barometer-a web accessibility evaluation and analysis platform. 2015.

[19] K. Sarita, P. Kaur, and S. Kaur. Accessibility of healthcare sites: evaluation by automated tools. In *Proceedings of International Conference on Data Science and Applications: ICDSA 2021, Volume 2*, pages 625–636. Springer, 2022.

[20] Scrapy developers. Architecture overview. https://docs.scrapy.org/en/latest/topics/architecture.html, 2024.

[21] Selenium developers. The selenium browser automation project. https://www.selenium.dev/documentation/, 2024.

[22] C. Sik-Lanyi and É. Orbán-Mihálykó. Accessibility testing of european health-related websites. *Arabian Journal for Science and Engineering*, 44(11):9171–9190, 2019.

[23] D. M. Swallow, H. Petrie, and C. D. Power. Understanding and supporting web developers:: design and evaluation of a web accessibility information resource (webair). In *Universal Design 2016: Learning from the past, designing for the future (Proceedings of the 3rd International Conference on Universal Design, UD2016)*, pages 482–491. IOS Press, 2016.

[24] M. Vigo, J. Brown, and V. Conway. Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In *Proceedings of the 10th international cross-disciplinary conference on web accessibility*, pages 1–10, 2013.

[25] K. White, E. Velleman, M. Hansma, and V. Lange. Selecting web accessibility evaluation tools. https://www.w3.org/WAI/test-evaluate/tools/selecting/, 2023.

[26] World Wide Web Consortium. Web content accessibility guidelines (WCAG) 2.0. https://www.w3.org/TR/WCAG20/, 2008.

[27] World Wide Web Consortium. WCAG 3 introduction. https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/, 2023.

[28] World Wide Web Consortium. Understanding conformance. https://www.w3.org/WAI/WCAG22/Understanding/conformance, 2024.

[29] World Wide Web Consortium. WCAG 2 overview. https://www.w3.org/WAI/standards-guidelines/wcag/, 2024.

[30] Y. Yesilada, G. Brajnik, and S. Harper. How much does expertise matter? a barrier walkthrough study with experts and non-experts. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pages 203–210, 2009.

Appendix A

Usability Questionnaire

Usability Survey

This survey is used to evaluate the usability of the Accessibility Evaluation website. The website can be accessed via http://martianmeggie.pythonanywhere.com.

* Required

Tasks

Please complete the tasks below.

1. For the overall analysis, which category has the most errors? *

- ◯ Operable
- ◯ Perceivable
- ◯ Robust
- ◯ Understandable

2. For the overall analysis, what are the IDs of the most common issues at each level (A, AA AAA)? *

- ◯ 117, 116, 38
- ◯ 48, 1, 7
- ◯ 174, 117, 306
- ◯ 213, 37, 223

3. For the district analysis, what is the average number of Errors in Liverpool? *

- ◯ 5.6
- ◯ 5.7
- ◯ 6.1
- ◯ 7.2

4. For the district analysis, how many districts have more than 40 mean Errors (bar can be dragged to set threshold)? *

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4

5. For the trend analysis, what is the percentage of Operable errors in 2025 Q2? *

- ◯ 17.04%
- ◯ 62.42%
- ◯ 21.04%
- ◯ 18.96%

6. For the trend analysis, which error category shows the most evident decreasing trend? *

- ◯ Operable
- ◯ Robust
- ◯ Understandable
- ◯ Perceivable

7. For the first GP (Ailsa Craig Medical Centre) listed in Manchester, how many potential problems does it have? *

- ◯ 23
- ◯ 7
- ◯ 10
- ◯ 15

8. Find a GP with postcode "M14 6WP", how many A level errors does it have? *

- ◯ 3
- ◯ 5
- ◯ 7
- ◯ 9

System Usability Scale

Please select the one you think is most suitable.
1-5: from strongly disagree to strongly agree.

9. I think that I would like to use this system frequently. *

1	2	3	4	5

Strongly disagree — Strongly agree

10. I found the system unnecessarily complex. *

1	2	3	4	5

Strongly disagree — Strongly agree

11. I thought the system was easy to use. *

1	2	3	4	5

Strongly disagree — Strongly agree

12. I think that I would need the support of a technical person to be able to use this system. *

1	2	3	4	5

Strongly disagree — Strongly agree

System Usability Scale

Please select the one you think is most suitable.
1-5: from strongly disagree to strongly agree.

9. I think that I would like to use this system frequently. *

1	2	3	4	5

Strongly disagree Strongly agree

10. I found the system unnecessarily complex. *

1	2	3	4	5

Strongly disagree Strongly agree

11. I thought the system was easy to use. *

1	2	3	4	5

Strongly disagree Strongly agree

12. I think that I would need the support of a technical person to be able to use this system. *

1	2	3	4	5

Strongly disagree Strongly agree

13. I found the various functions in this system were well integrated. *

1	2	3	4	5

Strongly disagree Strongly agree

14. I thought there was too much inconsistency in this system. *

1	2	3	4	5

Strongly disagree — Strongly agree

15. I would imagine that most people would learn to use this system very quickly. *

1	2	3	4	5

Strongly disagree — Strongly agree

16. I found the system very cumbersome to use. *

1	2	3	4	5

Strongly disagree — Strongly agree

17. I felt very confident using the system. *

1	2	3	4	5

Strongly disagree — Strongly agree

18. I needed to learn a lot of things before I could get going with this system. *

1	2	3	4	5

Strongly disagree — Strongly agree