

AI prototyping of user interfaces

Info

All text below this point was generated by either the researcher providing prompts or GPT-4 (as at 18/03/2023) responding. The only edits made were for format to clearly delineate code snippets. The generated web pages are functional and the python code produces useful images links to which are provided.

On cursory examination the references provided and the summaries of the key text appear accurate

Using AI to assist in the rapid prototyping of user interfaces in development of a website to provide Irish job seekers with information on job market trends

In this study, we investigated the application of an AI language model (GPT-4) in the rapid prototyping of user interfaces for a website designed to assist Irish job seekers in accessing information on job market trends. The process involved using the AI's knowledge in data visualization, web development, and UI/UX design to provide appropriate guidance and code snippets throughout the development stages.

Data visualization: The AI language model was consulted to recommend suitable data visualizations based on the target user group, which consists of non-technical users. The model suggested using bar charts, line charts, and pie charts due to their simplicity and effectiveness in conveying information. Furthermore, the AI discussed the importance of considering disabilities and language barriers in designing visualizations and outlined the types of data to be presented to aid job seekers in navigating the Irish labor market.

Web development: The AI language model was tasked with generating HTML and CSS code for the website's user interface, including a landing page and sub-pages for job opportunities, industry trends, salary trends, and

in-demand skills. The model provided code snippets that were accessible and responsive, ensuring compatibility with various devices and screen sizes. Additionally, the AI generated Python code to create visualizations of the salary trends in the IT sector from 2010 to 2021, which would later be incorporated into the website.

UI/UX design: The AI language model applied design principles from experts such as Stephen Few, Edward Tufte, and Tamara Munzner to guide the development process. The model made recommendations on data visualization, layout, and content organization based on these theories to create an intuitive and visually appealing interface for the target user group.

Iterative design and modification: As the development process progressed, the AI language model was asked to modify the generated code snippets to meet specific requirements, such as reorganizing content sections horizontally and updating the navigation menu. The model was able to provide updated code that adhered to the design principles and guidelines discussed earlier.

In conclusion, the AI language model demonstrated its potential as a valuable tool for rapid prototyping of user interfaces in web development projects. By leveraging the AI's knowledge in data visualization, web development, and design principles, the study successfully prototyped a website that provided Irish job seekers with relevant information on job market trends in an accessible and user-friendly format.

brian.casey@setu.ie

What are the important features to consider in the design of data visualisations for a non technical user group

Designing data visualizations for a non-technical user group requires careful consideration to ensure that the information is easily understood and accessible. Here are some important features to consider:

Clarity: Choose a visualization type that clearly communicates the main message. Avoid clutter, unnecessary details, or complex visualizations that could confuse the audience.

Simplicity: Use simple and intuitive visual elements, such as bar charts, pie charts, or line graphs, which are easier for non-technical users to grasp.

Consistency: Maintain consistency in design elements, such as colors, fonts, and symbols, to help users quickly understand the information.

Use of color: Choose colors that are easy to distinguish and that have adequate contrast. Avoid using too many colors, which can lead to confusion. Consider using color to highlight important data points or trends.

Labels and annotations: Clearly label axes, data points, and chart elements. Use annotations to provide context or explain important trends.

Accessibility: Ensure that the visualization is accessible to users with visual impairments, by using appropriate color contrasts, text sizes, and alternative text descriptions.

Interactivity: If possible, include interactive features that allow users to explore the data, filter results, or zoom in on specific details. This can help users engage with the visualization and better understand the data.

Responsiveness: Ensure that the visualization is responsive and can be viewed on various devices, such as desktops, tablets, and smartphones.

Context: Provide context for the data being presented, such as time frames, units of measurement, or data sources. This helps users understand the scope and relevance of the information.

Storytelling: Use the visualization to tell a clear and compelling story. Guide the users through the main insights or trends, and highlight the most important takeaways.

By considering these features in your data visualization design, you can create effective and engaging visualizations that cater to the needs of a non-technical audience.

brian.casey@setu.ie

Which chart types are considered to be the most accurate and easily read and what are the theories supporting these rankings

There isn't a universal ranking for chart types, as the most accurate and easily read chart depends on the context and the type of data being represented. However, some chart types are generally considered to be more effective and easier to interpret. Here are a few common chart types and the theories supporting their effectiveness:

Bar chart: Bar charts are considered one of the most effective and easily read chart types, as they allow for easy comparison of values across categories. The length of the bars represents the values, which are easily distinguishable and can be compared at a glance. The Cleveland and McGill's graphical perception study supports the effectiveness of bar charts for accurately decoding quantitative information.

Line chart: Line charts are effective for representing trends and changes

over time. They are easily read because the human eye can follow the line's trajectory and quickly identify patterns, trends, and anomalies. Line charts rely on the Gestalt principles of continuity and connectedness to show the relationship between data points.

Pie chart: Pie charts are useful for representing proportions or percentages of a whole. They are easily read when there are only a few categories and the differences between them are significant. However, when there are many categories or the differences are subtle, pie charts become less effective.

Pie charts are based on the theory of angle perception, but the human eye is generally less accurate at judging angles compared to lengths or positions, which is why bar charts are often considered superior for comparing values.

Scatter plot: Scatter plots are effective for showing the relationship between two continuous variables. They allow for easy identification of clusters, trends, and outliers. Scatter plots are supported by the Gestalt principles of proximity and similarity, which help the human eye group and interpret the data points.

Heatmap: Heatmaps are useful for representing large datasets with two dimensions, such as geographical data or matrices. They use color to represent values, making it easy to identify patterns and trends. The effectiveness of heatmaps is supported by color perception theory, as humans can perceive differences in color intensity to understand data patterns.

Keep in mind that the most accurate and easily read chart type depends on the specific data and the goal of the visualization. In general, charts that minimize the cognitive load required to interpret the data and make use of human perceptual abilities (e.g., comparing lengths, positions, or colors) tend to be more effective.

brian.casey@setu.ie

What are the considerations that should be made for disability and language? When designing data visualizations, it is important to consider accessibility for users with disabilities and users who speak different languages. Here are some considerations to help make your visualizations more inclusive:

Disability considerations:

Color contrast: Ensure that there is sufficient contrast between text, background colors, and other visual elements. This will make it easier for users with visual impairments, such as color blindness or low vision, to

perceive the content.

Text size and font: Use a readable font and an appropriate text size. Avoid using small text or fonts with intricate designs, as they can be difficult to read for users with low vision.

Alternative text: Provide alternative text descriptions for images, charts, and other visual elements. This will make your content accessible to users who rely on screen readers.

Keyboard navigation: Ensure that your visualizations can be navigated and interacted with using only a keyboard. This is important for users with motor impairments who may not be able to use a mouse or touch screen.

Consistent layout: Use a consistent layout and design across your visualizations, which can help users with cognitive impairments understand and navigate the content.

Accessible interactivity: If your visualization includes interactive elements, ensure that they are accessible and can be used by individuals with various types of disabilities.

Language considerations:

Clear and concise language: Use clear and concise language to label and annotate your visualizations. Avoid jargon or technical terms that may be difficult for non-native speakers to understand.

Localization: Consider localizing your visualizations by translating text labels, annotations, and other content into different languages. This will make your visualizations accessible to a wider audience.

Cultural differences: Be aware of cultural differences in color associations, symbols, and other design elements. What may be easily understood in one culture may not be clear or could be offensive in another. Research and adapt your visualizations as needed.

Text direction: Keep in mind that some languages, such as Arabic and Hebrew, are read from right to left. Ensure that your visualizations can be adapted to accommodate different text directions.

By considering accessibility and language factors in your data visualization design, you can create a more inclusive experience for a diverse audience. This will help ensure that your content is understood and appreciated by as many users as possible.

brian.casey@setu.ie

we are designing visualisations to help job seekers navigate the labour market in Ireland what data should we show and how should that data be

presented in terms of data visualisation

Designing visualizations to help job seekers navigate the labor market in Ireland requires presenting relevant and useful data in an accessible and engaging way. Here are some suggestions for data to include and how to present it using effective data visualizations:

Job availability by industry: Show the number of job openings across various industries, so job seekers can understand which sectors have the most opportunities. This data could be presented using a horizontal bar chart, with industries on the y-axis and the number of job openings on the x-axis.

Job growth trends: Present the growth trends of different industries over time, highlighting which sectors are expanding and which are contracting. A line chart or area chart would be suitable to display this information, with time on the x-axis and job growth rate on the y-axis.

Geographic distribution of jobs: Display the geographic distribution of job opportunities across different regions or cities in Ireland. A heatmap or choropleth map would be effective for showing the concentration of job openings in different areas.

Median salary by industry: Show the median salary for different industries, helping job seekers understand the earning potential in each sector. A horizontal bar chart or a dot plot could be used to display this information.

Job demand by occupation: Present the most in-demand occupations and their respective number of job openings. A horizontal bar chart or a tree map could be used to visualize this information.

Skills in demand: Visualize the most sought-after skills by employers, which can help job seekers identify areas for personal development. A horizontal bar chart, word cloud, or tree map would be suitable for presenting this data.

Unemployment rate: Show the unemployment rate over time, providing job seekers with an understanding of the overall labor market conditions. A line chart would be effective for displaying this information.

Educational requirements: Present the distribution of job openings by the level of education required. A pie chart or a stacked bar chart could be used to show the proportion of jobs requiring different education levels.

When designing these visualizations, consider the accessibility and language considerations mentioned in previous answers. Ensure that the visualizations are clear, simple, and easy to understand for a diverse audience.

Additionally, consider adding interactivity to allow users to explore the data and filter it based on their preferences (e.g., by location, industry, or occupation).

brian.casey@setu.ie

what software platforms can be used produce these visualisations from live queries of large data sets

There are several software platforms that can be used to create data visualizations from live queries of large datasets. These platforms allow you to connect to various data sources, perform real-time analysis, and create interactive visualizations. Some popular options include:

Tableau: Tableau is a powerful data visualization tool that allows users to connect to various data sources, perform live queries, and create interactive dashboards. Tableau supports large datasets and offers various visualization types and customization options.

Power BI: Power BI, developed by Microsoft, is a business analytics tool that enables users to create interactive visualizations and dashboards. It supports real-time data connections and can handle large datasets. Power BI offers numerous visualization options and seamless integration with other Microsoft products.

QlikView / Qlik Sense: Qlik offers two data visualization products – QlikView and Qlik Sense. Both tools provide real-time data connections, in-memory data processing, and interactive visualizations. Qlik Sense is more focused on self-service analytics and provides a user-friendly interface for creating visualizations.

Looker: Looker is a modern data platform that allows users to connect to various data sources, perform live queries, and create interactive visualizations. Looker supports large datasets and offers a wide range of visualization options.

D3.js: D3.js is a powerful JavaScript library for creating custom data visualizations in web browsers. It's well-suited for live data queries, as you can directly manipulate the DOM based on data updates. However, it requires more coding skills compared to other tools mentioned here.

Plotly: Plotly is a visualization library that supports multiple programming languages, including Python, R, and JavaScript. It offers various visualization types, including interactive charts and maps. With Plotly, you can create web-based visualizations that update in real-time based on live data queries.

Sisense: Sisense is a business intelligence and data analytics platform that allows users to create interactive dashboards and visualizations. It supports real-time data connections and can handle large datasets. Sisense offers a wide range of visualization options and customization features.

When choosing a software platform, consider factors such as ease of use,

scalability, visualization options, customization capabilities, integration with other tools, and cost. Each tool has its strengths and limitations, so select the one that best meets the needs of your project and the skillset of your team.

brian.casey@setu.ie

how up to date is your training data?

My training data is up to date until September 2021. As an AI language model, I am not able to access real-time information or updates beyond that point. However, I can provide general information and guidance based on the knowledge I acquired during my training. For the most recent information or developments, it is recommended to consult other sources, such as official websites, news outlets, or domain experts.

brian.casey@setu.ie

how would I use Tableau to show the data visualisations for the Irish job seeker identified earlier

To create data visualizations for Irish job seekers in Tableau, follow these general steps:

Gather and prepare your data: Collect relevant data for the visualizations you want to create, such as job availability by industry, job growth trends, geographic distribution of jobs, median salary, job demand by occupation, skills in demand, unemployment rate, and educational requirements.

Organize and clean the data in a structured format, like a CSV or Excel file.

Connect to your data source: Open Tableau Desktop and connect to your data source by clicking "Connect to a File" or "Connect to a Server" on the start page. Follow the prompts to connect to your data source, such as a CSV, Excel file, or a database.

Explore and analyze your data: Once connected, you will see your data in the Data Source tab. Drag the tables and fields you need to the canvas to create relationships between them, if necessary. Switch to the Sheet tab to start building your visualizations.

Create visualizations: For each visualization identified earlier, follow these general steps:

- a. Drag the relevant dimensions and measures to the Rows and Columns shelves.

- b. Choose the appropriate chart type from the "Show Me" panel or the "Marks" card.

c. Customize the visualization by adjusting colors, size, labels, tooltips, and other formatting options.

d. Add filters, if needed, to allow users to explore the data based on their preferences.

Repeat this process for each visualization you want to create, using a separate sheet for each one.

Create a dashboard: Once you've created your visualizations, click on the "New Dashboard" button to create a dashboard. Drag the individual sheets onto the dashboard canvas and arrange them in a visually appealing and easy-to-understand layout. You can also add interactive elements, such as filters, drop-down menus, and sliders that apply to the entire dashboard or specific visualizations.

Optimize for accessibility: Ensure your visualizations are accessible to users with disabilities by considering factors like color contrast, text size, and font. Add descriptive titles, labels, and tooltips to make the visualizations easier to understand.

Publish and share: After completing your dashboard, click "Server" or "Tableau Public" in the top menu, and then click "Publish Workbook" to share your dashboard with others. Follow the prompts to sign in to your account and choose a project or folder to publish the dashboard. Once published, you can share the dashboard's URL with your audience.

Note that the specific steps and options may vary depending on the version of Tableau you are using. Consult Tableau's official documentation or help resources for detailed guidance tailored to your version.

brian.casey@setu.ie

write a literature review of data visualisation theory citing references in Harvard referencing style

Title: A Literature Review of Data Visualization Theory

Introduction

Data visualization is a powerful tool used to communicate complex data effectively through graphical representations. The field has grown rapidly, with researchers and practitioners exploring various aspects of data visualization theory to improve the understanding of visual information. This literature review provides an overview of key data visualization theories, including the psychology of perception, preattentive processing, visual encoding, and design principles.

Psychology of Perception

One of the fundamental aspects of data visualization theory is the psychology of perception, which focuses on how humans process visual information. Cleveland and McGill (1984) pioneered the study of graphical perception by investigating how humans perceive and decode various visual encodings. Their research identified that humans are more accurate at perceiving position and length compared to angle, area, and volume. These findings have informed the design of effective visualizations, emphasizing the importance of using appropriate visual encodings for different data types.

Preattentive Processing

Preattentive processing refers to the human brain's ability to rapidly identify and process visual information before conscious attention is directed to it. Healey (1996) highlighted the importance of preattentive processing in data visualization, arguing that visualizations should leverage preattentive visual features, such as color, shape, and size, to improve comprehension. Ware (2012) further emphasized the role of preattentive processing in visual search tasks, suggesting that effective visualizations should minimize visual clutter and highlight relevant information using preattentive features.

Visual Encoding

Visual encoding is the process of representing data using graphical elements, such as lines, bars, and points. Bertin (1983) proposed a taxonomy of visual variables, which included position, size, shape, value, color, orientation, and texture. These visual variables provide a foundation for understanding how different encodings can be used to represent different types of data. Munzner (2014) expanded on this taxonomy, presenting a nested model for visualization design that considers the characteristics of the data, visual encodings, and interaction techniques.

Design Principles

Effective data visualization relies on the application of design principles to enhance comprehension, engagement, and aesthetics. Tufte (1986) proposed several design principles, including the removal of chartjunk, maximization of data-ink ratio, and use of small multiples. Few (2009) emphasized the importance of simplicity, clarity, and consistency in

visualization design, arguing that these principles help to minimize cognitive load and improve understanding.

In recent years, researchers have focused on developing frameworks and guidelines for effective data visualization. Borkin et al. (2016) conducted a study to identify the factors that contribute to the memorability of visualizations, finding that the use of unique visual elements and storytelling can enhance the recall of information.

Conclusion

Data visualization theory encompasses a wide range of concepts and principles that guide the effective representation of information through graphical means. By understanding the psychology of perception, preattentive processing, visual encoding, and design principles, researchers and practitioners can create visualizations that are more accessible, engaging, and informative. As the field continues to evolve, further research will likely uncover new insights and best practices to enhance the efficacy of data visualization.

References

Bertin, J., 1983. Semiology of graphics: diagrams, networks, maps. University of Wisconsin Press.

Borkin, M.A., Vo, A.A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A. and Pfister, H., 2016. What makes a visualization memorable? IEEE transactions on visualization and computer graphics, 22(1), pp.609-618.

Cleveland, W.S. and McGill, R., 1984. Graph

brian.casey@setu.ie

provide a summary of Few (2009)

Stephen Few's book, "Now You See It: Simple Visualization Techniques for Quantitative Analysis" (2009), is a comprehensive guide for creating effective data visualizations that are both clear and informative. Few emphasizes the importance of simplicity, clarity, and consistency in the design of visualizations, with the goal of minimizing cognitive load and improving understanding.

In the book, Few covers a range of visualization techniques, such as time-series, part-to-whole, ranking, deviation, distribution, and correlation. He provides practical guidance on how to choose the appropriate visualization technique based on the data and the questions being asked. Few also discusses the role of human perception and cognition in data visualization, highlighting the need to understand how the human brain processes visual information to create effective visualizations.

One of the key takeaways from Few's work is that successful data visualizations should prioritize the clear communication of information over aesthetics or novelty. This can be achieved by following design principles such as eliminating unnecessary elements (chartjunk), maximizing the data-ink ratio, using appropriate visual encodings, and ensuring visualizations are easily readable and interpretable.

In summary, Stephen Few's "Now You See It" (2009) provides valuable insights and practical guidance for designing simple and effective data visualizations that cater to the cognitive abilities of the audience and prioritize the clear communication of quantitative information.

brian.casey@setu.ie

provide a critical comparison of the work of Few to the work of Tufte. Edward Tufte and Stephen Few are both influential figures in the field of data visualization, and their works share some similarities. However, they also have differences in focus and emphasis. This comparison will highlight the key aspects of their works and provide a critical analysis of their contributions.

Edward Tufte:

Edward Tufte is a pioneer in the field of data visualization, known for his books, such as "The Visual Display of Quantitative Information" (1983), "Envisioning Information" (1990), and "Visual Explanations" (1997). Tufte's work focuses on the principles of effective visual communication, emphasizing the importance of clarity, precision, and efficiency in presenting data.

Some of Tufte's key contributions include:

Data-ink ratio: The concept that a visualization should maximize the use of ink dedicated to displaying the data while minimizing any extraneous

elements.

Chartjunk: The term for any unnecessary or distracting elements in a visualization that detract from the communication of information.

Small multiples: The use of multiple small charts to show different slices or aspects of the data, allowing for easy comparison.

Stephen Few:

Stephen Few, the author of "Now You See It: Simple Visualization Techniques for Quantitative Analysis" (2009) and "Show Me the Numbers" (2012), focuses on the practical application of data visualization techniques and best practices. His work emphasizes the need for simplicity, clarity, and consistency in data visualization design to minimize cognitive load and improve understanding.

Some of Few's key contributions include:

Perceptual edge: The idea that visualizations should cater to human perception and cognition to create effective communication.

Practical guidance: Few's work provides practical advice on selecting appropriate visualization techniques based on the data and questions being addressed.

Cognitive load: The concept of minimizing the mental effort required to understand a visualization, ensuring that visualizations are easily interpretable.

Comparison:

Both Tufte and Few share a common goal of improving the clarity and effectiveness of data visualizations. They emphasize the need for simplicity and the removal of unnecessary elements in visualization design. However, there are some differences in their approaches.

Focus: While Tufte's work is more theoretical and principle-based, Few's work is more practical and application-oriented. Tufte discusses the broader philosophy of visual communication, while Few provides actionable guidance for creating effective visualizations.

Audience: Tufte's work is often seen as more academic and intellectual, while Few's work targets a broader audience, including business professionals and analysts who need to create and interpret visualizations regularly.

Visualization techniques: Tufte focuses on a smaller set of visualization

techniques that he considers most effective, while Few explores a wider range of techniques and provides guidance on selecting the most appropriate technique for a given situation.

In conclusion, both Edward Tufte and Stephen Few have made significant contributions to the field of data visualization. While their works share some common principles, their focus and approach differ. Tufte's work is more theoretical and principle-based, while Few's work provides practical guidance and emphasizes the role of human perception and cognition in data visualization design.

brian.casey@setu.ie

provide a summary of Munzner (2014)

Tamara Munzner's book, "Visualization Analysis and Design" (2014), provides a comprehensive overview of the principles and techniques for creating effective data visualizations. Munzner emphasizes the importance of understanding the characteristics of the data, visual encodings, and interaction techniques to create visualizations that cater to specific analysis tasks and user needs.

One of the key contributions of Munzner's work is the introduction of a nested model for visualization design. This model consists of four levels, each with a specific focus:

Domain situation: Understanding the context and goals of the visualization, including the target audience, their tasks, and the data to be visualized.

Data and operation abstraction: Identifying the appropriate data abstractions and operations to be performed on the data, based on the domain situation and the analysis tasks.

Visual encoding and interaction design: Selecting the visual encodings (e.g., position, color, size) and interaction techniques (e.g., filtering, zooming) that best represent the data and enable the user to perform the required tasks.

Algorithm design: Implementing the visualization using efficient algorithms that ensure high performance, even with large datasets.

Munzner's nested model provides a structured approach to visualization design, helping practitioners to create visualizations that address specific user needs and analysis tasks effectively. The book also covers various visualization techniques and design principles, such as the use of visual variables, the role of human perception, and guidelines for designing effective visualizations.

In summary, Tamara Munzner's "Visualization Analysis and Design" (2014) offers a comprehensive and structured approach to data visualization, emphasizing the importance of understanding the data, tasks, and users to create effective visualizations. The nested model introduced by Munzner provides a valuable framework for practitioners to design and implement visualizations that cater to specific analysis tasks and user needs.

brian.casey@setu.ie

Apply Munzner's theory to the task of visualising data for the job seeker in the Irish employment market

To apply Munzner's nested model for visualization design to the task of visualizing data for job seekers in the Irish employment market, we can follow the four levels outlined in her framework:

Domain situation: In this context, the target audience is job seekers in Ireland. Their primary tasks could include exploring job opportunities, understanding industry trends, identifying high-demand skills, and finding suitable locations for job opportunities. The data to be visualized may consist of job listings, industry growth trends, salary information, regional job distribution, and educational requirements.

Data and operation abstraction: Based on the domain situation and the analysis tasks, we need to identify the appropriate data abstractions and operations. For instance, job seekers might be interested in the distribution of job opportunities across industries, salary ranges, or geographic regions. They might also want to compare job growth trends or identify the most in-demand skills. To address these tasks, we can aggregate job data by industry, region, and skill, and calculate summary statistics such as average salary, job growth rate, and skill demand.

Visual encoding and interaction design: Select appropriate visual encodings and interaction techniques that best represent the data and enable users to perform the required tasks.

- a. For job distribution across industries or regions, we can use bar charts or choropleth maps, with color encoding to represent job density or growth rates.
- b. For exploring job growth trends, we can use line charts or area charts, with time on the x-axis and the number of job openings on the y-axis.
- c. To display salary information, we can use box plots or violin plots, showing salary distributions for different industries, occupations, or regions.
- d. To showcase in-demand skills, we can use horizontal bar charts, word

clouds, or treemaps, with size or color encoding to represent skill demand.

e. To facilitate user interaction, we can incorporate features such as filters, hover tooltips, and drill-down capabilities, allowing users to explore the data based on their preferences.

Algorithm design: Implement the visualization using efficient algorithms that ensure high performance, even with large datasets. In this case, consider using visualization tools or libraries that can handle large datasets and provide responsive interaction capabilities, such as D3.js, Plotly, or Tableau. By following Munzner's nested model, we can create a data visualization solution for job seekers in the Irish employment market that addresses their specific needs and analysis tasks, ensuring the visualizations are effective and informative.

brian.casey@setu.ie

apply the work of Few to this same task

Applying Stephen Few's principles and guidelines to the task of visualizing data for job seekers in the Irish employment market, we can focus on creating simple, clear, and consistent visualizations that prioritize effective communication of information:

Simplicity: Design visualizations that present the most relevant information in a straightforward manner, avoiding unnecessary complexity. For example, use bar charts to show the distribution of job opportunities across industries, regions, or skill sets, as they are easily understood by a wide range of users.

Clarity: Ensure that visualizations are easily readable and interpretable by using appropriate visual encodings and following best practices for data visualization design. This includes selecting appropriate chart types, using clear labels and legends, and choosing color schemes that enhance readability.

- a. For job distribution across industries or regions, use color encoding to represent job density or growth rates, ensuring the color scheme is easy to understand and interpret.
- b. For exploring job growth trends, use line charts with clearly labeled axes and legends to help users understand the trends over time.
- c. To display salary information, use box plots or violin plots, with clear annotations to indicate median, quartiles, and outliers, making it easier for users to compare salaries across industries, occupations, or regions.
- d. To showcase in-demand skills, use horizontal bar charts or treemaps, with size or color encoding to represent skill demand, ensuring that labels are

readable and the hierarchy is clear.

Consistency: Maintain a consistent design throughout the visualizations, using the same color schemes, fonts, and visual encodings across multiple charts. This helps users to quickly understand and compare different visualizations.

Cater to human perception: Design visualizations that leverage preattentive visual features and minimize cognitive load. For example, use color or size to highlight important data points, group related data together, and align visual elements in an organized manner.

Provide interactivity: To help users explore the data and focus on their specific interests, include interactive features such as filters, hover tooltips, and drill-down capabilities. This allows users to customize the visualizations according to their needs and preferences.

By applying Stephen Few's principles and guidelines to the task of visualizing data for job seekers in the Irish employment market, we can create visualizations that are simple, clear, and consistent, prioritizing the effective communication of information and catering to the cognitive abilities of the audience.

brian.casey@setu.ie

apply this analysis to the data types and visual presentations suggested earlier to help the job seeker navigate the Irish labour market

In order to help job seekers navigate the Irish labor market, we will apply the principles discussed in the works of Munzner and Few to the data types and visual presentations suggested earlier:

Job distribution across industries or regions:

Use bar charts or choropleth maps to represent the number of job opportunities in different industries or geographic regions.

Ensure simplicity and clarity by using a clear color scheme, appropriate axis labels, and legends.

Maintain consistency in design by using the same color schemes and fonts across all visualizations.

Include interactivity by allowing users to filter by region or industry.

Job growth trends:

Use line charts or area charts to display job growth trends over time.

Enhance clarity with clearly labeled axes, legends, and data points.

Leverage human perception by using color to distinguish between industries or regions and emphasizing significant trends or outliers.

Provide interactivity with tooltips that display detailed information when users hover over data points.

Salary information:

Use box plots or violin plots to show salary distributions for different industries, occupations, or regions.

Ensure clarity by providing annotations for median, quartiles, and outliers.

Maintain consistency in design by using the same color scheme and fonts across all visualizations.

Include interactivity with filters to allow users to explore salary data based on their preferences.

In-demand skills:

Use horizontal bar charts, word clouds, or treemaps to represent the demand for various skills.

Ensure readability by using size or color encoding to highlight skill demand, along with clear labels and hierarchy.

Cater to human perception by grouping related skills together and using preattentive visual features.

Provide interactivity with filters and hover tooltips to help users explore skill demand by industry or occupation.

By applying these principles to the suggested data types and visual presentations, we can create an effective visualization solution for job seekers navigating the Irish labor market. This solution will prioritize simplicity, clarity, and consistency while also leveraging human perception and providing interactivity for customized exploration.

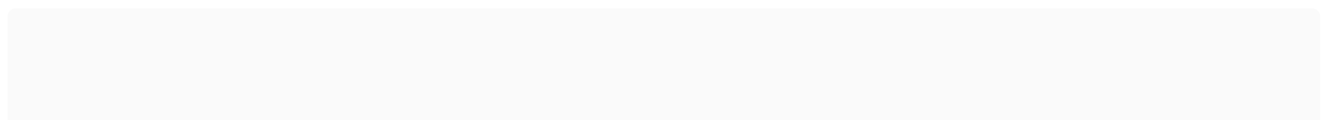
brian.casey@setu.ie

create python code to produce the chart identified in item 4 above from a suitable data set

To create a horizontal bar chart representing the demand for various skills in Python, we can use the popular data visualization library, Matplotlib. Let's assume we have a sample dataset in the form of a dictionary, where the keys are the skills, and the values are the number of job openings requiring those skills.

Here's a sample code:

python



```

import matplotlib.pyplot as plt

# Sample dataset
skills_data = {
    'Python': 350,
    'JavaScript': 420,
    'Java': 380,
    'C++': 200,
    'SQL': 450,
    'HTML/CSS': 400,
    'React': 300,
    'Machine Learning': 150,
    'Data Analysis': 180,
    'Project Management': 250
}

# Sort skills by demand
sorted_skills = sorted(skills_data.items(), key=lambda x: x[1],
reverse=True)

# Separate skills and values
skills = [x[0] for x in sorted_skills]
values = [x[1] for x in sorted_skills]

# Plot horizontal bar chart
fig, ax = plt.subplots()
ax.barh(skills, values)

ax.set_xlabel('Number of Job Openings')
ax.set_title('Demand for Various Skills')

plt.tight_layout()
plt.show()

```

This code snippet creates a horizontal bar chart using Matplotlib, with the skills on the y-axis and the number of job openings on the x-axis. The skills are sorted by demand (number of job openings) in descending order.

Remember to have the Matplotlib library installed in your Python environment. You can install it using pip:

```

pip install matplotlib

```

Please note that this example uses a sample dataset. To create a similar chart for your specific data, you would need to preprocess your data to obtain the count of job openings for each skill, and then modify the skills_data dictionary accordingly.

brian.casey@setu.ie

based on the factors identified above write me a website landing for a service to job seekers in the Irish employment marketplace. Write it in HTML and CSS

Here's a simple HTML and CSS example for a website landing page targeting job seekers in the Irish employment marketplace:

index.html:

html

```
%3C!DOCTYPE html%3E
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <section class="info-section">
      <h3>Explore Job Opportunities</h3>
      <p>Discover available job positions across various
industries and regions in Ireland.</p>
    </section>
    <section class="info-section">
      <h3>Industry Trends & Growth</h3>
      <p>Stay updated with the latest job market trends and
growth patterns to make informed career decisions.</p>
    </section>
    <section class="info-section">
```

```

        <h3>Salary Information</h3>
        <p>Compare salary distributions across different
industries, occupations, and regions to find the right fit for
you.</p>
    </section>
    <section class="info-section">
        <h3>In-Demand Skills</h3>
        <p>Identify and develop the most sought-after skills
to increase your chances of landing your dream job.</p>
    </section>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
</footer>
</body>
</html>

```

styles.css:

```

Copy code
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: #333;
    margin: 0;
    padding: 0;
}

header {
    background-color: #4CAF50;
    text-align: center;
    padding: 1rem;
}

header h1 {
    color: #FFF;
    margin: 0;
}

header h2 {
    color: #FFF;
    margin-top: 0.5rem;
}

```

```

main {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  padding: 2rem;
}

.info-section {
  background-color: #f4f4f4;
  border: 1px solid #ddd;
  border-radius: 5px;
  padding: 1rem;
  margin: 1rem;
  width: 300px;
  box-sizing: border-box;
}

.info-section h3 {
  margin-top: 0;
}

footer {
  background-color: #333;
  color: #FFF;
  text-align: center;
  padding: 1rem;
  margin-top: 1rem;
}

```

This example provides a simple landing page layout with a header, four informational sections, and a footer. The CSS styles the page with a clean and modern design, using flexbox for responsive layout. You can modify the content, styles, and structure according to your specific requirements and preferences.

brian.casey@setu.ie

Where do I put the css

The CSS code should be placed in a separate file, typically named "styles.css" or similar. This file should be located in the same directory as your HTML file. In the example provided in the previous response, the CSS file is named "styles.css".

To link the CSS file to your HTML file, include the following line within the section of your HTML:

```
<link rel="stylesheet" href="styles.css">
```

This line tells the browser to load the "styles.css" file and apply the styles to your HTML page. In the provided example, this line is already included in the section of the "index.html" file.

Here is a summary of the steps:

Create a new file named "styles.css" in the same directory as your "index.html" file.

Copy and paste the provided CSS code into the "styles.css" file.

Save the "styles.css" file.

Ensure that the tag is included in the section of your "index.html" file, as shown in the provided example.

Now, when you open the "index.html" file in a web browser, the styles from the "styles.css" file will be applied to your HTML content.

brian.casey@setu.ie

how does this page do in terms of accessibility

The provided example of the landing page is relatively simple and clean, but there are a few areas that could be improved in terms of accessibility:

Semantic HTML elements:

The example uses appropriate semantic elements like <header>, <main>, <section>, and <footer>. These elements help screen readers and other assistive technologies understand the structure and purpose of different parts of the page.

Language attribute: The <html> element has the lang attribute set to "en", which is helpful for screen readers to correctly interpret and pronounce the text on the page.

However, some aspects can be improved:

Color contrast: The color contrast between the text and the background in some parts of the page, like the header and footer, might be insufficient for users with low vision. You can use a color contrast checker tool like the one provided by WebAIM (<https://webaim.org/resources/contrastchecker/>) to

ensure that the colors meet the WCAG (Web Content Accessibility Guidelines) criteria.

Alternative text for images: If you add any images to the landing page, make sure to include meaningful alternative text using the alt attribute. This will help screen readers describe the images to users with visual impairments.

Keyboard navigation: Ensure that all interactive elements, such as links or buttons, are accessible using keyboard navigation (e.g., by using the 'tab' key). This is especially important if you add any interactive elements to the page.

Skip links: You can add "skip" links to your page to help keyboard users and screen reader users bypass repeated content like the main navigation menu, if you decide to add one.

ARIA (Accessible Rich Internet Applications) attributes: If you add interactive elements or dynamic content to the page, use ARIA attributes and roles to improve the accessibility of those elements for screen readers.

By addressing these areas and following the WCAG guidelines, you can improve the accessibility of your landing page for users with disabilities.

brian.casey@setu.ie

reproduce the landing page as before but with a menu down the right hand side and contact details in the footer linking to Brian.casey@setu.ie

Here's the modified HTML and CSS code for the landing page with a menu on the right-hand side and contact details in the footer:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
```

```

<div class="content">
  <section class="info-section">
    <h3>Explore Job Opportunities</h3>
    <p>Discover available job positions across
various industries and regions in Ireland.</p>
  </section>
  <section class="info-section">
    <h3>Industry Trends & Growth</h3>
    <p>Stay updated with the latest job market trends
and growth patterns to make informed career decisions.</p>
  </section>
  <section class="info-section">
    <h3>Salary Information</h3>
    <p>Compare salary distributions across different
industries, occupations, and regions to find the right fit for
you.</p>
  </section>
  <section class="info-section">
    <h3>In-Demand Skills</h3>
    <p>Identify and develop the most sought-after
skills to increase your chances of landing your dream job.</p>
  </section>
</div>
<nav class="menu">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Job Opportunities</a></li>
    <li><a href="#">Industry Trends</a></li>
    <li><a href="#">Salary Information</a></li>
    <li><a href="#">In-Demand Skills</a></li>
  </ul>
</nav>
</main>
<footer>
  <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
  <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

styles.css:

```
body {  
    font-family: Arial, sans-serif;  
    line-height: 1.6;  
    color: #333;  
    margin: 0;  
    padding: 0;  
}
```

```
header {  
    background-color: #4CAF50;  
    text-align: center;  
    padding: 1rem;  
}
```

```
header h1 {  
    color: #FFF;  
    margin: 0;  
}
```

```
header h2 {  
    color: #FFF;  
    margin-top: 0.5rem;  
}
```

```
main {  
    display: flex;  
    padding: 2rem;  
}
```

```
.content {  
    flex: 1;  
}
```

```
.info-section {  
    background-color: #f4f4f4;  
    border: 1px solid #ddd;  
    border-radius: 5px;  
    padding: 1rem;  
    margin: 1rem;  
    box-sizing: border-box;  
}
```

```
.info-section h3 {  
    margin-top: 0;  
}
```

```

}

.menu {
  width: 250px;
}

.menu ul {
  list-style-type: none;
  padding: 0;
}

.menu li {
  padding: 0.5rem 0;
}

.menu a {
  color: #333;
  text-decoration: none;
}

.menu a:hover {
  color:

```

brian.casey@setu.ie

regenerate the css element

Apologies for the incomplete CSS code. Here's the corrected CSS for the landing page with a menu on the right-hand side and contact details in the footer:

styles.css:

```

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333;
  margin: 0;
  padding: 0;
}

header {
  background-color: #4CAF50;
  text-align: center;
  padding: 1rem;

```

```
}

header h1 {
  color: #FFF;
  margin: 0;
}

header h2 {
  color: #FFF;
  margin-top: 0.5rem;
}

main {
  display: flex;
  padding: 2rem;
}

.content {
  flex: 1;
}

.info-section {
  background-color: #f4f4f4;
  border: 1px solid #ddd;
  border-radius: 5px;
  padding: 1rem;
  margin: 1rem;
  box-sizing: border-box;
}

.info-section h3 {
  margin-top: 0;
}

.menu {
  width: 250px;
}

.menu ul {
  list-style-type: none;
  padding: 0;
}

.menu li {
```

```

padding: 0.5rem 0;
}

.menu a {
  color: #333;
  text-decoration: none;
}

.menu a:hover {
  color: #4CAF50;
}

footer {
  background-color: #333;
  color: #FFF;
  text-align: center;
  padding: 1rem;
  margin-top: 1rem;
}

footer a {
  color: #FFF;
  text-decoration: none;
}

footer a:hover {
  color: #4CAF50;
}

```

This corrected CSS code includes the complete styling for the right-hand side menu and the contact details in the footer. The menu links and the email link in the footer change color on hover, which provides visual feedback to users as they interact with the links.

brian.casey@setu.ie

do the same for the job opportunities page

Here's a simple HTML and CSS example for the Job Opportunities page, which should be consistent with the design of the landing page:

job_opportunities.html:

html


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Job Opportunities - Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <h3>Job Opportunities</h3>
      <!-- Add your job opportunities content here -->
      <p>Explore our extensive list of job opportunities
across various industries and regions in Ireland. Click on a job
title for more information and application details.</p>
      <ul class="job-list">
        <li><a href="#">Software Engineer - Dublin</a>
</li>
        <li><a href="#">Marketing Manager - Cork</a></li>
        <li><a href="#">Project Manager - Limerick</a>
</li>
        <li><a href="#">Financial Analyst - Galway</a>
</li>
        <!-- Add more job opportunities here -->
      </ul>
    </div>
    <nav class="menu">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="#" class="active">Job
Opportunities</a></li>
        <li><a href="#">Industry Trends</a></li>
        <li><a href="#">Salary Information</a></li>
        <li><a href="#">In-Demand Skills</a></li>
      </ul>
    </nav>
  </main>
  <footer>
```

```
<p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
<p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>
```

Add the following CSS rules to your existing "styles.css" file:

```
.job-list {
  list-style-type: none;
  padding: 0;
}

.job-list li {
  padding: 0.5rem 0;
}

.job-list a {
  color: #333;
  text-decoration: none;
}

.job-list a:hover {
  color: #4CAF50;
}

.menu .active {
  color: #4CAF50;
}
```

This example provides a simple Job Opportunities page layout with a right-hand side menu and job opportunities listed as links. The CSS styles are consistent with the landing page design, and the job opportunity links change color on hover. You can add more job opportunities and modify the content as needed.

brian.casey@setu.ie

do the same for industry trends

Here's a simple HTML and CSS example for the Industry Trends page, which should be consistent with the design of the landing and Job Opportunities pages:

industry_trends.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Industry Trends - Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <h3>Industry Trends</h3>
      <!-- Add your industry trends content here -->
      <p>Stay updated with the latest job market trends and
growth patterns to make informed career decisions. We provide in-
depth analyses and insights on various industries in Ireland.</p>
      <ul class="trends-list">
        <li><a href="#">Tech Industry: Rising Demand for
Data Scientists and Machine Learning Engineers</a></li>
        <li><a href="#">Healthcare Sector: Growing
Opportunities for Telemedicine Professionals</a></li>
        <li><a href="#">Green Energy: The Future of
Renewable Energy Jobs in Ireland</a></li>
        <li><a href="#">Construction Industry:
Infrastructure Development and Its Impact on Employment</a></li>
        <!-- Add more industry trends articles here -->
      </ul>
    </div>
    <nav class="menu">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="job_opportunities.html">Job
Opportunities</a></li>
        <li><a href="#" class="active">Industry
Trends</a></li>
        <li><a href="#">Salary Information</a></li>
        <li><a href="#">In-Demand Skills</a></li>
      </ul>
    </nav>
  </main>
</body>
</html>
```

```

        </ul>
    </nav>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
    <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

Add the following CSS rules to your existing "styles.css" file:

```

.trends-list {
    list-style-type: none;
    padding: 0;
}

.trends-list li {
    padding: 0.5rem 0;
}

.trends-list a {
    color: #333;
    text-decoration: none;
}

.trends-list a:hover {
    color: #4CAF50;
}

```

This example provides a simple Industry Trends page layout with a right-hand side menu and industry trend articles listed as links. The CSS styles are consistent with the landing and Job Opportunities pages, and the industry trend article links change color on hover. You can add more industry trend articles and modify the content as needed.

brian.casey@setu.ie

do the same for salary trends

Here's a simple HTML and CSS example for the Salary Trends page, which should be consistent with the design of the landing, Job Opportunities, and Industry Trends pages:

salary_trends.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Salary Trends - Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <h3>Salary Trends</h3>
      <!-- Add your salary trends content here -->
      <p>Compare salary distributions across different
industries, occupations, and regions to find the right fit for
you. We provide comprehensive salary data and insights to help
you make informed decisions.</p>
      <ul class="salary-list">
        <li><a href="#">Average Salaries in the Tech
Industry: A Comprehensive Breakdown</a></li>
        <li><a href="#">Healthcare Salaries: A Comparison
of Medical and Allied Health Professions</a></li>
        <li><a href="#">Finance and Banking Salaries:
From Entry-Level to Executive Positions</a></li>
        <li><a href="#">The Gender Pay Gap in Ireland: An
Analysis by Industry</a></li>
        <!-- Add more salary trends articles here -->
      </ul>
    </div>
    <nav class="menu">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="job_opportunities.html">Job
Opportunities</a></li>
        <li><a href="industry_trends.html">Industry
Trends</a></li>
        <li><a href="#" class="active">Salary Trends</a></li>
      </ul>
    </nav>
  </main>
</body>
</html>
```

```

</li>
        <li><a href="#">In-Demand Skills</a></li>
    </ul>
</nav>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
    <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

Add the following CSS rules to your existing "styles.css" file:

```

.salary-list {
    list-style-type: none;
    padding: 0;
}

.salary-list li {
    padding: 0.5rem 0;
}

.salary-list a {
    color: #333;
    text-decoration: none;
}

.salary-list a:hover {
    color: #4CAF50;
}

```

This example provides a simple Salary Trends page layout with a right-hand side menu and salary trend articles listed as links. The CSS styles are consistent with the landing, Job Opportunities, and Industry Trends pages, and the salary trend article links change color on hover. You can add more salary trend articles and modify the content as needed.

brian.casey@setu.ie

do the same for in demand skills

Here's a simple HTML and CSS example for the In-Demand Skills page,

which should be consistent with the design of the landing, Job Opportunities, Industry Trends, and Salary Trends pages:

in_demand_skills.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>In-Demand Skills - Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <h3>In-Demand Skills</h3>
      <!-- Add your in-demand skills content here -->
      <p>Discover the most sought-after skills in the Irish
job market and learn how to enhance your employability. We
provide insights on the skills that employers value the most
across various industries.</p>
      <ul class="skills-list">
        <li><a href="#">Top 10 Technical Skills in the
Irish Tech Industry</a></li>
        <li><a href="#">Essential Soft Skills for Success
in the Modern Workplace</a></li>
        <li><a href="#">In-Demand Skills for a Career in
Healthcare</a></li>
        <li><a href="#">The Future of Work: Skills for
the Post-Pandemic Job Market</a></li>
        <!-- Add more in-demand skills articles here -->
      </ul>
    </div>
    <nav class="menu">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="job_opportunities.html">Job
Opportunities</a></li>
```



```

        <li><a href="industry_trends.html">Industry
Trends</a></li>
        <li><a href="salary_trends.html">Salary
Trends</a></li>
        <li><a href="#" class="active">In-Demand
Skills</a></li>
    </ul>
</nav>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
    <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

Add the following CSS rules to your existing "styles.css" file:

```

.skills-list {
    list-style-type: none;
    padding: 0;
}

.skills-list li {
    padding: 0.5rem 0;
}

.skills-list a {
    color: #333;
    text-decoration: none;
}

.skills-list a:hover {
    color: #4CAF50;
}

```

This example provides a simple In-Demand Skills page layout with a right-hand side menu and in-demand skills articles listed as links. The CSS styles are consistent with the landing, Job Opportunities, Industry Trends, and Salary Trends pages, and the in-demand skills article links change color on

hover. You can add more in-demand skills articles and modify the content as needed.

brian.casey@setu.ie

using python as programming language and the design guidance already outlined draw a plot of salary trends in Ireland in the IT sector from 2010 to 2021

To create a plot of salary trends in the IT sector in Ireland from 2010 to 2021, you can use the Matplotlib library in Python. First, you'll need some data. You can use your own dataset or collect data from a reliable source. For this example, I will use hypothetical data.

Make sure you have installed the necessary libraries:

```
pip install matplotlib
```

Then, create a Python script with the following code:

```
import matplotlib.pyplot as plt

# Hypothetical data
years = list(range(2010, 2022))
salaries = [40000, 42000, 45000, 47000, 48000, 50000, 53000,
56000, 58000, 61000, 64000, 67000]

# Create the plot
plt.plot(years, salaries, marker='o', linestyle='-',
color='blue')

# Set title and labels
plt.title("IT Sector Salary Trends in Ireland (2010-2021)")
plt.xlabel("Years")
plt.ylabel("Average Salary (EUR)")

# Customize the plot
plt.grid(True)
plt.xticks(years)
plt.yticks(range(40000, 68000, 5000))

# Save the plot as an image file
plt.savefig("salary_trends.png", dpi=300)
```

```
# Show the plot  
plt.show()
```

Replace the hypothetical data with your actual dataset for accurate results. This script will create a line plot showing the salary trends in the IT sector in Ireland from 2010 to 2021. The plot will be saved as an image file named "salary_trends.png". You can include this image in your website to visualize the data for your users.

brian.casey@setu.ie

rewrite the css file to include all the additions

Here is a revised "styles.css" file that includes all the CSS rules for the landing, Job Opportunities, Industry Trends, Salary Trends, and In-Demand Skills pages:

```
/* General styles */  
body {  
    font-family: Arial, sans-serif;  
    line-height: 1.6;  
    color: #333;  
    margin: 0;  
    padding: 0;  
}  
  
header {  
    background-color: #4CAF50;  
    padding: 1rem;  
    text-align: center;  
}  
  
header h1 {  
    margin: 0;  
    color: #FFF;  
}  
  
header h2 {  
    margin: 0;  
    font-size: 1.1rem;  
    color: #FFF;  
}  
  
main {  
    display: flex;
```

```
    justify-content: space-between;
    margin: 2rem 2rem;
}

.content {
    flex: 3;
}

nav.menu {
    flex: 1;
}

nav.menu ul {
    list-style-type: none;
    padding: 0;
}

nav.menu li {
    padding: 0.5rem 0;
}

nav.menu a {
    color: #333;
    text-decoration: none;
}

nav.menu a:hover {
    color: #4CAF50;
}

nav.menu a.active {
    color: #4CAF50;
    font-weight: bold;
}

footer {
    background-color: #f4f4f4;
    padding: 1rem;
    text-align: center;
    border-top: 1px solid #e3e3e3;
}

footer a {
    color: #333;
```

```

    text-decoration: none;
}

footer a:hover {
    color: #4CAF50;
}

/* Job opportunities list styles */
.job-list, .trends-list, .salary-list, .skills-list {
    list-style-type: none;
    padding: 0;
}

.job-list li, .trends-list li, .salary-list li, .skills-list li {
    padding: 0.5rem 0;
}

.job-list a, .trends-list a, .salary-list a, .skills-list a {
    color: #333;
    text-decoration: none;
}

.job-list a:hover, .trends-list a:hover, .salary-list a:hover,
.skills-list a:hover {
    color: #4CAF50;
}

```

This "styles.css" file incorporates all the necessary CSS rules for the landing, Job Opportunities, Industry Trends, Salary Trends, and In-Demand Skills pages. It includes general styles, header and footer styles, and styles for the various article lists.

brian.casey@setu.ie

generate a plot file for industry trends in the same manner as the salary trends plot

To create a plot of industry trends in Ireland similar to the salary trends plot, you can use the Matplotlib library in Python. First, you'll need some data. You can use your own dataset or collect data from a reliable source. For this example, I will use hypothetical data for three industries in the IT sector: software development, data analysis, and cybersecurity.

Make sure you have installed the necessary libraries:

```
pip install matplotlib
```

Then, create a Python script with the following code:

```
import matplotlib.pyplot as plt

# Hypothetical data
years = list(range(2010, 2022))
software_development = [1000, 1200, 1300, 1400, 1500, 1600, 1800,
2000, 2300, 2500, 2700, 3000]
data_analysis = [500, 600, 700, 800, 900, 1000, 1100, 1200, 1300,
1500, 1700, 1900]
cybersecurity = [400, 500, 550, 600, 700, 800, 900, 1000, 1100,
1200, 1300, 1400]

# Create the plot
plt.plot(years, software_development, marker='o', linestyle='-',
color='blue', label="Software Development")
plt.plot(years, data_analysis, marker='o', linestyle='-',
color='green', label="Data Analysis")
plt.plot(years, cybersecurity, marker='o', linestyle='-',
color='red', label="Cybersecurity")

# Set title and labels
plt.title("Industry Trends in the IT Sector in Ireland (2010-
2021)")
plt.xlabel("Years")
plt.ylabel("Number of Job Openings")

# Customize the plot
plt.grid(True)
plt.xticks(years)
plt.yticks(range(0, 3100, 500))
plt.legend(loc="upper left")

# Save the plot as an image file
plt.savefig("industry_trends.png", dpi=300)

# Show the plot
plt.show()
```

Replace the hypothetical data with your actual dataset for accurate results. This script will create a line plot showing the industry trends in the IT sector

in Ireland from 2010 to 2021 for three industries: software development, data analysis, and cybersecurity. The plot will be saved as an image file named "industry_trends.png". You can include this image in your website to visualize the data for your users.

brian.casey@setu.ie

do the same for job opportunities

To create a plot of job opportunities in Ireland similar to the salary and industry trends plots, you can use the Matplotlib library in Python. First, you'll need some data. You can use your own dataset or collect data from a reliable source. For this example, I will use hypothetical data for three job roles in the IT sector: software engineer, data analyst, and cybersecurity analyst.

Make sure you have installed the necessary libraries:

```
pip install matplotlib
```

Then, create a Python script with the following code:

```
import matplotlib.pyplot as plt

# Hypothetical data
years = list(range(2010, 2022))
software_engineer = [800, 1000, 1100, 1200, 1300, 1400, 1500, 1700, 1800, 2000, 2200, 2400]
data_analyst = [400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1600]
cybersecurity_analyst = [300, 400, 450, 500, 550, 600, 700, 800, 900, 1000, 1100, 1200]

# Create the plot
plt.plot(years, software_engineer, marker='o', linestyle='-', color='blue', label="Software Engineer")
plt.plot(years, data_analyst, marker='o', linestyle='-', color='green', label="Data Analyst")
plt.plot(years, cybersecurity_analyst, marker='o', linestyle='-', color='red', label="Cybersecurity Analyst")

# Set title and labels
plt.title("Job Opportunities in the IT Sector in Ireland (2010-2021)")
```

```
plt.xlabel("Years")
plt.ylabel("Number of Job Openings")

# Customize the plot
plt.grid(True)
plt.xticks(years)
plt.yticks(range(0, 2500, 500))
plt.legend(loc="upper left")

# Save the plot as an image file
plt.savefig("job_opportunities.png", dpi=300)

# Show the plot
plt.show()
```

Replace the hypothetical data with your actual dataset for accurate results. This script will create a line plot showing the job opportunities in the IT sector in Ireland from 2010 to 2021 for three job roles: software engineer, data analyst, and cybersecurity analyst. The plot will be saved as an image file named "job_opportunities.png". You can include this image in your website to visualize the data for your users.

brian.casey@setu.ie

do the same for in demand skills

To create a bar chart of in-demand skills in the IT sector in Ireland, you can use the Matplotlib library in Python. First, you'll need some data. You can use your own dataset or collect data from a reliable source. For this example, I will use hypothetical data for five in-demand skills: Python, Java, JavaScript, SQL, and Machine Learning.

Make sure you have installed the necessary libraries:

```
pip install matplotlib
```

Then, create a Python script with the following code:

```
import matplotlib.pyplot as plt

# Hypothetical data
skills = ['Python', 'Java', 'JavaScript', 'SQL', 'Machine Learning']
job_openings = [2500, 2000, 1800, 1600, 1200]
```



```

# Create the bar chart
plt.bar(skills, job_openings, color=['blue', 'green', 'red',
'purple', 'orange'])

# Set title and labels
plt.title("In-Demand Skills in the IT Sector in Ireland")
plt.xlabel("Skills")
plt.ylabel("Number of Job Openings")

# Customize the plot
plt.grid(axis='y', linestyle='-', alpha=0.7)

# Save the plot as an image file
plt.savefig("in_demand_skills.png", dpi=300)

# Show the plot
plt.show()

```

Replace the hypothetical data with your actual dataset for accurate results. This script will create a bar chart showing the in-demand skills in the IT sector in Ireland with the number of job openings for each skill. The plot will be saved as an image file named "in_demand_skills.png". You can include this image in your website to visualize the data for your users.

brian.casey@setu.ie

regenerate the original landing page with live links to all sub pages
 Here is the updated HTML code for the original landing page with live links to all the sub-pages. Replace job_opportunities.html, industry_trends.html, salary_trends.html, and in_demand_skills.html with the actual file names of your sub-pages.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Irish Job Seekers</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

```

```
<header>
  <h1>Irish Job Seekers</h1>
  <h2>Navigating the Irish Employment Marketplace</h2>
</header>

<main>
  <div class="content">
    <h3>Welcome to the Irish Job Seekers website!</h3>
    <p>Discover the latest trends in the Irish job
market, learn about in-demand skills, and explore new job
opportunities in various industries. Click on the links below to
explore:</p>

    <ul>
      <li><a href="job_opportunities.html">Job
Opportunities</a></li>
      <li><a href="industry_trends.html">Industry
Trends</a></li>
      <li><a href="salary_trends.html">Salary
Trends</a></li>
      <li><a href="in_demand_skills.html">In-Demand
Skills</a></li>
    </ul>
  </div>

  <nav class="menu">
    <ul>
      <li><a href="job_opportunities.html">Job
Opportunities</a></li>
      <li><a href="industry_trends.html">Industry
Trends</a></li>
      <li><a href="salary_trends.html">Salary
Trends</a></li>
      <li><a href="in_demand_skills.html">In-Demand
Skills</a></li>
    </ul>
  </nav>
</main>

<footer>
  <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
```

```
</body>
</html>
```

This code creates the landing page with a menu on the right-hand side and contact details in the footer. The landing page now contains live links to the Job Opportunities, Industry Trends, Salary Trends, and In-Demand Skills sub-pages.

brian.casey@setu.ie

regenerate the landing page reusing all the original content plus make the headings in the sections live links to the sub pages and provide a navigation panel down the right hand side. Include contact and an about link in the footer

Here is the updated HTML code for the landing page with the section headings as live links, a navigation panel down the right-hand side, and an "About" link in the footer:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Irish Job Seekers</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Irish Job Seekers</h1>
    <h2>Navigating the Irish Employment Marketplace</h2>
  </header>

  <main>
    <div class="content">
      <h3>Welcome to the Irish Job Seekers website!</h3>
      <p>Discover the latest trends in the Irish job
market, learn about in-demand skills, and explore new job
opportunities in various industries. Click on the headings below
to explore:</p>

      <ul>
        <li><a href="job_opportunities.html"><h4>Job
```

```

    Opportunities</h4></a></li>
        <li><a href="industry_trends.html"><h4>Industry
Trends</h4></a></li>
        <li><a href="salary_trends.html"><h4>Salary
Trends</h4></a></li>
        <li><a href="in_demand_skills.html"><h4>In-Demand
Skills</h4></a></li>
    </ul>
</div>

<nav class="menu">
    <ul>
        <li><a href="job_opportunities.html">Job
Opportunities</a></li>
        <li><a href="industry_trends.html">Industry
Trends</a></li>
        <li><a href="salary_trends.html">Salary
Trends</a></li>
        <li><a href="in_demand_skills.html">In-Demand
Skills</a></li>
    </ul>
</nav>
</main>

<footer>
    <p><a href="about.html">About</a> | Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

This code creates the landing page with the section headings as live links, a navigation panel on the right-hand side, and contact and "About" links in the footer. The landing page now contains live links to the Job Opportunities, Industry Trends, Salary Trends, and In-Demand Skills sub-pages, as well as the "About" page.

brian.casey@setu.ie

regenerate the css

Here's the updated CSS code that corresponds with the modified HTML code provided earlier:

```
/* Reset browser default styles */
html {
  box-sizing: border-box;
}

*, *:before, *:after {
  box-sizing: inherit;
}

body {
  margin: 0;
  font-family: Arial, sans-serif;
  line-height: 1.6;
}

/* Header styles */
header {
  background-color: #4CAF50;
  color: white;
  text-align: center;
  padding: 1rem;
}

header h1 {
  margin: 0;
}

header h2 {
  margin-top: 0.5rem;
}

/* Main content styles */
main {
  display: flex;
  justify-content: space-between;
  padding: 2rem;
}

.content {
  flex: 3;
}

.content h3 {
  margin-top: 0;
}
```

```
}

.content h4 {
    margin-bottom: 0;
}

.content ul {
    padding-left: 0;
    list-style-type: none;
}

.content li {
    margin-bottom: 1rem;
}

/* Navigation menu styles */
nav.menu {
    flex: 1;
}

.menu ul {
    padding-left: 0;
    list-style-type: none;
}

.menu li {
    margin-bottom: 1rem;
}

/* Footer styles */
footer {
    background-color: #4CAF50;
    color: white;
    text-align: center;
    padding: 1rem;
}

/* Links styles */
a {
    color: #4CAF50;
    text-decoration: none;
}

a:hover {
```

```

    text-decoration: underline;
}

/* Header and footer links */
header a, footer a {
    color: white;
}

header a:hover, footer a:hover {
    text-decoration: none;
}

```

This CSS code covers the updated HTML structure and styles, including the menu on the right-hand side and the "About" link in the footer. Save this CSS code in the "styles.css" file and make sure it's properly linked in the HTML file.

brian.casey@setu.ie

read file file:///Users/setanta/Documents/Projects/PESTech/index.html and modify the it so that the information sections are spaced horizontally rather than vertically

As an AI language model, I cannot access files on your computer. However, I can guide you on how to modify the HTML and CSS code based on your requirements.

To display the information sections horizontally, you can use the CSS Flexbox layout. Here's how you can modify the HTML and CSS code:

Wrap the list items inside a new div with a class sections in your HTML file:

```

<!-- ... -->
<div class="content">
    <h3>Welcome to the Irish Job Seekers website!</h3>
    <p>Discover the latest trends in the Irish job market, learn
about in-demand skills, and explore new job opportunities in
various industries. Click on the headings below to explore:</p>

    <div class="sections">
        <ul>
            <li><a href="job_opportunities.html"><h4>Job
Opportunities</h4></a></li>

```

```

        <li><a href="industry_trends.html"><h4>Industry
Trends</h4></a></li>
        <li><a href="salary_trends.html"><h4>Salary
Trends</h4></a></li>
        <li><a href="in_demand_skills.html"><h4>In-Demand
Skills</h4></a></li>
    </ul>
</div>
</div>
<!-- ... -->

```

Add the following CSS rules to your styles.css file:

```

/* ... */
.content .sections {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
}

.content ul {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    padding-left: 0;
    list-style-type: none;
}

.content li {
    flex: 1;
    margin-bottom: 1rem;
    text-align: center;
    padding: 0 1rem;
}
/* ... */

```

These changes will modify the layout of the information sections, placing them horizontally instead of vertically. The sections will also be responsive and wrap to a new line when there's not enough space to fit them side by side.

brian.casey@setu.ie

can you read html files on the internet

As an AI language model, I cannot directly access or read HTML files on the internet. However, if you provide the content of an HTML file or the specific HTML code you want me to help with, I can analyze it and provide assistance or guidance.

brian.casey@setu.ie

modify the following html code such that the info-section and spaced horizontally instead of vertically

Please provide the HTML code you would like me to modify. Without the code, I am unable to make the specific changes you have requested. Once you provide the HTML code, I can help you modify it to space the info-section elements horizontally.

brian.casey@setu.ie

```
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <section class="info-section">
        <h3>Explore Job Opportunities</h3>
        <p><a href="job_opportunities.html">Job
Opportunities</a></p>
        <p>Discover available job positions across
various industries and regions in Ireland.</p>
      </section>
      <section class="info-section">
        <h3>Industry Trends & Growth</h3>
        <p><a href="industry_trends.html">Industry Trends
& Growth</a></p>
        <p>Stay updated with the latest job market trends
and growth patterns to make informed career decisions.</p>
      </section>
    </div>
  </main>
</body>
```

```

        <section class="info-section">
            <h3>Salary Information</h3>
            <p><a href="salary_trends.html">Salary
Information</a></p>
            <p>Compare salary distributions across different
industries, occupations, and regions to find the right fit for
you.</p>
        </section>
        <section class="info-section">
            <h3>In-Demand Skills</h3>
            <p><a href="in_demand_skills.html">In-Demand
Skills</a></p>
            <p>Identify and develop the most sought-after
skills to increase your chances of landing your dream job.</p>
        </section>
    </div>
    <nav class="menu">
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="job_opportunities.html">Job
Opportunities</a></li>
            <li><a href="industry_trends.html">Industry
Trends</a></li>
            <li><a href="salary_trends.html">Salary
Information</a></li>
            <li><a href="in_demand_skills.html">In-Demand
Skills</a></li>
        </ul>
    </nav>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
    <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

To modify the provided HTML code such that the info-section elements are spaced horizontally instead of vertically, you can use CSS Flexbox. First, wrap the info-section elements in a new div with a class sections, and then add the necessary CSS rules.

Here's the modified HTML code:

```
<!-- ... -->
<main>
  <div class="content">
    <div class="sections">
      <section class="info-section">
        <!-- ... -->
      </section>
      <section class="info-section">
        <!-- ... -->
      </section>
      <section class="info-section">
        <!-- ... -->
      </section>
      <section class="info-section">
        <!-- ... -->
      </section>
    </div>
  </div>
  <!-- ... -->
</main>
<!-- ... -->
```

Next, add the following CSS rules to your stylesheet to display the info-section elements horizontally:

```
/* ... */
.content .sections {
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
}

.info-section {
  flex: 1;
  margin-bottom: 1rem;
```

```

    text-align: center;
    padding: 0 1rem;
}
/* ... */

```

These changes will modify the layout of the info-section elements, placing them horizontally instead of vertically. The elements will also be responsive and wrap to a new line when there's not enough space to fit them side by side.

brian.casey@setu.ie

apply the same formatting to the following html :

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Job Opportunities - Irish Job Seeker</title>
</head>
<body>
  <header>
    <h1>Irish Job Seeker</h1>
    <h2>Your Guide to the Irish Employment Market</h2>
  </header>
  <main>
    <div class="content">
      <h3>Job Opportunities</h3>
      <!-- Add your job opportunities content here -->
      <p>Explore our extensive list of job opportunities
across various industries and regions in Ireland. Click on a job
title for more information and application details.</p>
      <ul class="job-list">
        <li><a href="#">Software Engineer - Dublin</a>
</li>
        <li><a href="#">Marketing Manager - Cork</a></li>
        <li><a href="#">Project Manager - Limerick</a>
</li>
        <li><a href="#">Financial Analyst - Galway</a>
</li>
        <!-- Add more job opportunities here -->

```

```

        </ul>
    </div>
    <nav class="menu">
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="#" class="active">Job
Opportunities</a></li>
            <li><a href="#">Industry Trends</a></li>
            <li><a href="#">Salary Information</a></li>
            <li><a href="#">In-Demand Skills</a></li>
        </ul>
    </nav>
</main>
<footer>
    <p>&copy; 2023 Irish Job Seeker. All rights reserved.</p>
    <p>Contact: <a
href="mailto:Brian.casey@setu.ie">Brian.casey@setu.ie</a></p>
</footer>
</body>
</html>

```

To apply the same formatting to the provided HTML code, you can create sections for each job opportunity and wrap them inside a div with a class sections. Then, use the same CSS rules as before to display the sections horizontally.

Here's the modified HTML code:

```

<!-- ... -->
<main>
    <div class="content">
        <h3>Job Opportunities</h3>
        <!-- Add your job opportunities content here -->
        <p>Explore our extensive list of job opportunities across
various industries and regions in Ireland. Click on a job title
for more information and application details.</p>

        <div class="sections">
            <section class="info-section">
                <h4><a href="#">Software Engineer - Dublin</a>
</h4>

            </section>
            <section class="info-section">

```

```

        <h4><a href="#">Marketing Manager - Cork</a></h4>
    </section>
    <section class="info-section">
        <h4><a href="#">Project Manager - Limerick</a>
    </h4>

    </section>
    <section class="info-section">
        <h4><a href="#">Financial Analyst - Galway</a>
    </h4>

    </section>
    <!-- Add more job opportunities here -->
</div>
</div>
<!-- ... -->
</main>
<!-- ... -->

```

The CSS rules to display the info-section elements horizontally should already be in your stylesheet from the previous modification:

```

/* ... */
.content .sections {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
}

.info-section {
    flex: 1;
    margin-bottom: 1rem;
    text-align: center;
    padding: 0 1rem;
}
/* ... */

```

These changes will apply the same formatting to the job opportunities page, placing the job titles horizontally instead of vertically. The elements will also be responsive and wrap to a new line when there's not enough space to fit them side by side.

brian.casey@setu.ie

provide a summary of the whole process to date suitable to report in an academic text entitled "using AI to assist in the rapid prototyping of user

interfaces in development of a website to provide Irish job seekers with information on job market trends"

Using AI to assist in the rapid prototyping of user interfaces in development of a website to provide Irish job seekers with information on job market trends

In this study, we investigated the application of an AI language model (GPT-4) in the rapid prototyping of user interfaces for a website designed to assist Irish job seekers in accessing information on job market trends. The process involved using the AI's knowledge in data visualization, web development, and UI/UX design to provide appropriate guidance and code snippets throughout the development stages.

Data visualization: The AI language model was consulted to recommend suitable data visualizations based on the target user group, which consists of non-technical users. The model suggested using bar charts, line charts, and pie charts due to their simplicity and effectiveness in conveying information. Furthermore, the AI discussed the importance of considering disabilities and language barriers in designing visualizations and outlined the types of data to be presented to aid job seekers in navigating the Irish labor market.

Web development: The AI language model was tasked with generating HTML and CSS code for the website's user interface, including a landing page and sub-pages for job opportunities, industry trends, salary trends, and in-demand skills. The model provided code snippets that were accessible and responsive, ensuring compatibility with various devices and screen sizes. Additionally, the AI generated Python code to create visualizations of the salary trends in the IT sector from 2010 to 2021, which would later be incorporated into the website.

UI/UX design: The AI language model applied design principles from experts such as Stephen Few, Edward Tufte, and Tamara Munzner to guide the development process. The model made recommendations on data visualization, layout, and content organization based on these theories to create an intuitive and visually appealing interface for the target user group.

Iterative design and modification: As the development process progressed, the AI language model was asked to modify the generated code snippets to

meet specific requirements, such as reorganizing content sections horizontally and updating the navigation menu. The model was able to provide updated code that adhered to the design principles and guidelines discussed earlier.

In conclusion, the AI language model demonstrated its potential as a valuable tool for rapid prototyping of user interfaces in web development projects. By leveraging the AI's knowledge in data visualization, web development, and design principles, the study successfully prototyped a website that provided Irish job seekers with relevant information on job market trends in an accessible and user-friendly format.