Developing an Assistive Education Tool for Data Visualization

by

Jai Narula

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2024 by the
Graduate Supervisory Committee:

Chris Bryan, Chair
Hasti Seifi
Srividya Bansal

ARIZONA STATE UNIVERSITY

May 2024

ABSTRACT

This research project seeks to develop an innovative data visualization tool tailored for beginners to enhance their ability to interpret and present data effectively. Central to the approach is creating an intuitive, user-friendly interface that simplifies the data visualization process, making it accessible even to those with no prior background in the field. The tool will introduce users to standard visualization formats and expose them to various alternative chart types, fostering a deeper understanding and broader skill set in data representation. I plan to leverage innovative visualization techniques to ensure the tool is compelling and engaging. An essential aspect of my research will involve conducting comprehensive user studies and surveys to assess the tool's impact on enhancing data visualization competencies among the target audience. Through this, I aim to gather valuable insights into the tool's usability and effectiveness, enabling further refinements. The outcome of this project is a powerful and versatile tool that will be an invaluable asset for students, researchers, and professionals who regularly engage with data. By democratizing data visualization skills, I envisage empowering a broader audience to comprehend and creatively present complex data in a more meaningful and impactful manner.

ACKNOWLEDGMENTS

The last two years have been an incredible rollercoaster of successes and setbacks on my path to earn a master's degree in computer science at ASU. When I think back on this time, I am proud that I was able to fulfill my main objective, which was to fully immerse myself in the huge ocean of information. This thesis is evidence of my unwavering pursuit of knowledge, and it would not have been possible without the steadfast assistance of many people who have enhanced this experience. With all my heart, I thank you.

Among my supporters, Dr. Chris Bryan is the most prominent. His data visualization course, which combined technical precision with the flexibility to explore creativity through visual narrative, was enlightening for me. I became interested in this topic due to Dr. Bryan's research and methods for approaching data visualization. I will always be indebted to him for accepting me as one of his thesis students and for giving me the freedom to follow my interests during our investigation. Dr. Bryan and his Ph.D. student Anjana Arunkumar gave essential laboratory mentorship. Through numerous cycles of brainstorming and revision, their knowledge and collaborative spirit were essential in turning my thesis into a work of major research worth.

My friends were just as important to this experience as I was. Their readiness to listen when you were frustrated or confused frequently opened the door to answers that did not appear to be in reach. Even when the subjects were beyond their expertise, their tolerance and understanding have been consoling and motivating. I am incredibly grateful to each of them for sharing this journey with me and for their company.

Finally, I would like to thank everyone who has supported me, given advice, and shared in my struggles and victories. Your group's encouragement has served as my compass, pointing me in the direction of my goals. I appreciate you being a part of my narrative.

TABLE OF CONTENTS

Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The teaching tools available in the rapidly developing field of data science and visualization frequently fail to meet the different learning demands and styles of pupils, especially those with impairments. The lack of easily available and user-friendly data visualization tools for novices is a major concern in the data science that the project directly addresses. The current landscape often leaves novices needing help to grasp the full potential of data visualization, potentially leading to subpar decision-making and analysis due to ineffective learning methods.

The project is focused on developing a user-friendly data visualization tool to tackle this difficulty. This tool is more than simply a piece of software; it is a thorough learning companion that was carefully created for people who are just starting out with data visualization. Understanding the daunting transition from inexperience to expertise, particularly for individuals without a background in data visualization, this application places a high value on usability and simplicity while retaining strong functionality. It will guide users through the fundamentals of data visualization with step-by-step tutorials and interactive experiences. We want to help people develop a deeper understanding so they can convey and analyze information visually. The goal is not just to make visual data representation easier.

The initiative's comprehensive review procedure is one of its core components. Following Mayer's multimedia learnings [1] guidelines, I plan to carry out in-depth human assessments to determine how this tool affects students fully determine how tool affects students. These assessments will not only determine and offer crucial input for improving user engagement and learning objectives. I hope to improve the product by using this evidence-based approach so that it better suits the users' educational needs and preferences.

Data visualization project's instructional potential is increased when an orchestration tool is added. Through interactive exercises and in-depth lectures, this tool will walk users through the fundamentals of data visualization and create a more dynamic and adaptable learning environment.

This disparity poses a serious obstacle to accessible learning in data visualization courses. The main goal of this project is to investigate how to best accommodate this variability in the design of an assistive teaching tool for data visualization. Three key questions that are intended to improve the learning process are at the center of the investigation:

- Facilitating Ease of Learning: What features or methodologies within assistive tools can simplify the learning process for users? This includes identifying the types of visual and interactive elements that can demystify complex data visualization concepts for learners with varying abilities. The focus will be on uncovering approaches that align with different cognitive styles and learning preferences, ensuring that the tool is inclusive and effective for a broad audience.
- Implementing Assistive Prompts: What kind of prompts or guided instructions can be integrated into the tool to assist students more effectively? This question delves

into the design of intuitive and adaptive prompts that can provide timely and context-specific guidance.

- Enhancing Interactivity for Engaging Learning: How can interactivity be leveraged to make the learning process more engaging and effective for users? Interactivity is pivotal in maintaining learner engagement and facilitating active learning. This part of the research will investigate the integration of interactive elements such as drag-and-drop interfaces, real-time data manipulation, and simulated scenarios that allow users to experiment and learn through doing.

Through tackling these inquiries, the study seeks to establish a foundation for creating an assisted education instrument that is inclusive, accessible, and efficient for students with a range of requirements and preferences. By giving all students access to a tool that enables them to succeed in the field of data visualization, the goal is to close the gap in today's educational practices.

1.2 Contributions

This thesis's goal is to develop a framework for explaining how students can be guided to harness the everyday potential of data visualization with an interactive visualization assistive tool. The goal is to investigate how these kinds of prompts can provide learning scaffolding—that is, step-by-step guidance that adapts to the user's needs and obstacles. These prompts could be anything from interactive walkthroughs designed to support students at different learning stages to textual suggestions and visual signals. The system's architectural design, which includes separate frontend and backend

components, attempts to provide users with an all-encompassing and dynamic learning experience by carrying out several crucial tasks:

- Task Initialization and Text Editor Provision: The system assigns students learning assignments that are thoughtfully crafted to stimulate and captivate them. These assignments come with an integrated text editor, a flexible tool that provides students with an easy-to-use interface to draft, modify, and turn in their work. This feature is essential for enabling a smooth connection between the lessons and the students' answers, enabling them to participate in the tasks at hand in real time.
- Admin-Driven Assistance and Feedback: Offering assistance and direction by administrators (or teachers) via a strong feedback system is another essential feature. By bridging the communication gap between students and teachers, this system component enables students to receive prompt feedback on their submissions and individualized help. It gives administrators the ability to give helpful criticism, recommendations for enhancements, and encouragement straight within the platform, improving the learning process and creating a positive learning environment.
- Centralized Monitoring and Management: Administrators will monitor the progress and work of all users within the system. This central overview allows for a more tailored and practical instructional approach. The tool will enable tracking user engagement and progression, which is crucial for understanding individual learning curves and areas requiring more attention.

Through the integration of these features, the system improves the quality of instruction provided while also streamlining the learning process. It is evidence of how technology has

the power to completely transform the way that educational content is accessed and comprehended, meeting the ever-changing demands of both educators and students. Finding out how these interactive elements might promote a more engaging and practical learning environment is the aim to improve understanding and memory of data visualization ideas.

By utilizing the system's potential, users received far more interactive feedback, which was essential for improving their comprehension of the ideas. The user-friendly feedback mechanism enhanced their learning experience by facilitating a discourse among users and providing clarification on intricate concepts.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Education theories in Data Visualization

Data visualization encompasses the sophisticated process of interpreting and deriving meaningful insights from datasets. At its core, this discipline leverages visual elements to simplify the complexity of data, making it accessible and understandable. Tables, a fundamental tool in data visualization, excel in presenting data points in a structured and organized manner. Their strength lies in their ability to display labels and quantities clearly, which becomes particularly potent when enhanced with sorting and filtering functionalities. These features unlock the tables' full potential, enabling users to navigate through and analyze data efficiently.

On the other hand, charts represent another pivotal aspect of data visualization, offering a dynamic approach to data interpretation. Charts transform data dimensions into visual properties of geometric shapes, creating a visual map that facilitates the comprehension of complex relationships and patterns within the data. This method of visualization not only aids in the analysis but also engages the viewer, fostering a deeper understanding of the data's underlying stories.

Both tables and charts serve as essential tools in the educational landscape of data visualization [2], embodying the principles of cognitive load theory and constructivist learning. They simplify the data analysis process, reduce cognitive overload, and support learners in constructing their understanding of data-driven insights. Through the strategic

use of these visualization tools, educators can enhance learners' ability to process, analyze, and extract meaningful information from data, illustrating the significant role of educational theories in shaping effective data visualization practices.

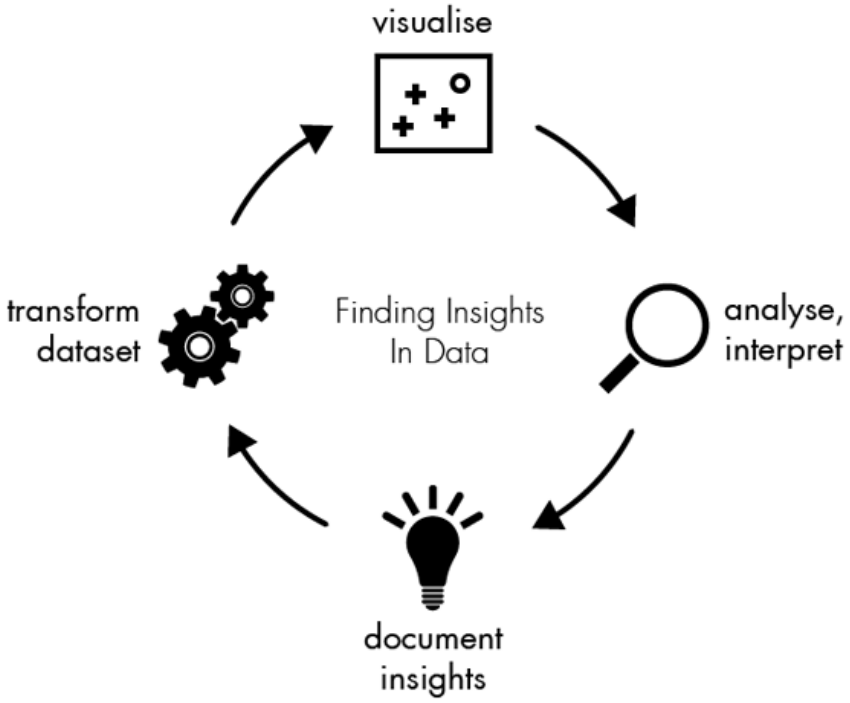


Figure 2.1 Data insights: a visualization (Gregor Aisch). Source: Data journalism [21]

2.2 Adaptive Learning Techniques

Using data-driven insights, adaptive learning is a sophisticated strategy that specifically tailors educational experiences to meet the needs of individual students. Adaptive learning[3] systems dynamically alter instructional content and methodology to provide highly personalized learning routes by carefully monitoring indicators like student progress, engagement levels, and overall performance.

Highlighted in McGuire's 2021 study, adaptive learning presents several key advantages:

- Enhanced Autonomy and Success: Adaptive learning systems empower students by providing clear insights into their current mastery levels, thereby facilitating a more autonomous learning journey. This self-paced approach allows learners to progress according to their individual capabilities and comprehension rates, potentially leading to higher levels of success.
- Increased Engagement: By delivering content and activities precisely aligned with each student's learning needs and preferences, adaptive learning significantly boosts student engagement. This personalized approach ensures that the learning material remains relevant and challenging enough to maintain interest without causing frustration.

These benefits collectively underscore the transformative potential of adaptive learning in creating more effective, engaging, and student-centered educational experiences.

2.3 Pedagogical Strategies for Teaching Data Visualization

Pedagogical strategies for teaching data visualization emphasize creating an interactive and engaging learning environment that caters to the diverse needs of learners. Key strategies include the integration of real-world projects to provide practical experience, the use of collaborative learning to foster peer-to-peer interaction, and the implementation of flipped classroom models to encourage active learning outside traditional lecture settings. Additionally, leveraging digital tools and platforms can facilitate interactive explorations of data sets, allowing students to practice data

manipulation and interpretation firsthand. Scaffolded learning experiences, where students progress from simple to more complex visualization tasks, can also help build confidence and skills incrementally. These strategies are supported by educational theories that advocate for experiential and collaborative learning, as seen in works by Dewey (1938) [4] and Vygotsky (1978) [5] and align with contemporary approaches to teaching complex cognitive skills, as discussed by Mayer (2009) [1] in the principles of multimedia learning.

In the realm of data visualization education, there is a tendency to prioritize instruction on methods rather than on the crucial aspect of communication. Despite widespread recognition of the importance of effective communication, the challenge lies in effectively teaching these skills, which often falls outside the comfort zones of many educators. The science of visualization, as elaborated in works like "The Grammar of Graphics" by Wilkinson [6], has been extensively explored, highlighting the structured approach to creating visual representations of data. There is a consensus among scholars and practitioners alike that equipping students with the ability to communicate through visualizations is of paramount importance. This shift towards emphasizing communication skills in the curriculum requires a reevaluation of pedagogical strategies, underscoring the need for educators to integrate comprehensive communication training in data visualization courses. This approach not only enhances the interpretative skills of students but also prepares them to convey complex data-driven insights more effectively in various professional contexts.

2.4 Existing tools for Data Visualization

The landscape of data visualization is rich with a variety of tools designed to cater to unique needs, skill levels, and purposes. These tools range from simple, user-friendly applications aimed at beginners and educators, to sophisticated platforms suitable for data scientists and professional analysts. Below is an overview of some key categories and examples of existing data visualization tools:

1. General-Purpose Visualization Tools:
    - Tableau [7]: Known for its ease of use and powerful interactive dashboards, Tableau is widely used in business and academia for creating complex visualizations.
    - Microsoft Power BI [8]: A suite of business analytics tools that deliver insights throughout an organization. It offers comprehensive data analysis capabilities with an emphasis on business intelligence.
    - Google Data Studio [9]: A free tool that turns data into customizable informative reports and dashboards. It is particularly user-friendly for beginners.
2. Programming-Based Tools:
    - R with ggplot2 [10]: R is a programming language dedicated to statistical computing and graphics, with ggplot2 being a widely acclaimed package for creating data visualizations.
    - Python with libraries such as Matplotlib and Seaborn [11][12]: Python is a versatile programming language, with Matplotlib and Seaborn being popular libraries for static, animated, and interactive visualizations.

- D3.js [13]: A JavaScript library for producing dynamic, interactive data visualizations in web browsers. It allows for the creation of highly customizable graphics.

3. Specialized Visualization Tools:
    - Gephi [14]: An open-source network analysis and visualization software package designed specifically for graphs and network data.
    - Qlik Sense [15]: A self-service data analysis and visualization tool that enables users to create personalized, interactive data visualizations.
4. Educational and Simplified Tools:
    - Datawrapper [16]: An online tool designed for making simple and effective charts and maps, widely used by journalists and educators.
    - Infogram [17]: A web-based visualization tool to create infographics, charts, and maps, emphasizing ease of use and visual appeal.

To meet the various needs of consumers in various sectors, each of these tools has specific features and functionalities. The requirements of the project, including the degree of customization needed, the complexity of the data, and the user's programming ability, are often the deciding factors when selecting a tool. These tools are updated frequently with new features and capabilities that reflect the most recent developments and trends in data visualization.

CHAPTER 3

THE INTERFACE

3.1 Backend

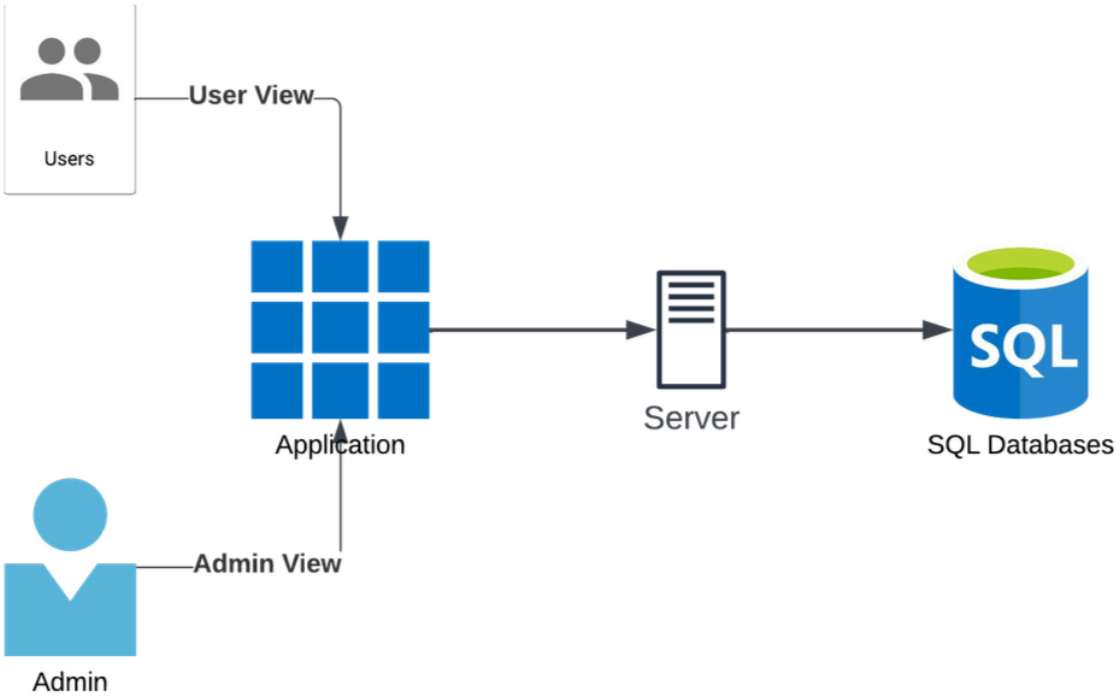


Figure 3.1.1 Architecture Diagram of the tool

The backend architecture, as illustrated in the provided diagram, is meticulously crafted utilizing the Django framework [18], a high-level Python web framework renowned for its efficiency in enabling rapid development coupled with a pragmatic and clean design philosophy.

1. User and Admin Interfaces: Central to the architecture are two principal interfaces: User View and Admin View. User View serves as the point of interaction for the

end-users, the learners who will be engaging with the data visualization tasks and activities facilitated by the tool. It is tailored to provide an intuitive and user-friendly experience, accommodating a diverse range of user capabilities, and ensuring that the educational content is accessible to all users, including those with disabilities. In parallel, Admin View is dedicated to administrators, allowing them to oversee and manage the system effectively. It enables tasks such as the initiation and modification of visualization activities, monitoring user progress, and providing personalized feedback. The Django framework's [18] inherent 'batteries-included' feature set supports the development of this robust admin interface, offering administrators a comprehensive dashboard for user management and content curation.

2. Application Logic: At the core of the architecture lies the Application itself, which embodies the logic and functionality of the educational tool. This Application is the operational heart of the system, responsible for handling and processing requests, executing core business logic, managing data transactions, and crafting appropriate responses. It is within this domain that the Django framework's capabilities shine, streamlining database schema migrations, user authentication processes, and URL routing.
3. Server and Database Interactions: The Server, depicted as the central node connecting the Application to the database, represents the physical or cloud-based server infrastructure hosting the Django application. This critical component ensures that the application remains responsive and accessible, managing the interplay between user actions and server-side processing.

4. Vegalite API: The Vega-Lite API [19] presents a streamlined approach to data visualization by leveraging a declarative syntax encapsulated within JSON objects. This declarative framework facilitates the specification of interactive visualizations without the need for verbose code, promoting a higher level of abstraction that simplifies the design and implementation process for developers and researchers. The JSON metadata, which acts as the blueprint for rendering charts, encompasses all necessary properties of visualization, from data sources to graphical mark types and encoding channels. In practice, the JSON object supplied to the Vega-Lite API defines the data set and instructs the library on how to map data attributes to visual properties, construct scales and axes, apply transformations and aggregations, and implement interactive features. Due to this metadata-driven architecture, Vega-Lite seamlessly integrates with various data processing pipelines, making it an ideal choice for applications that require dynamic and responsive chart rendering based on user interactions or real-time data updates.

```
vl.markBar().data('data/example.Json').encode(
  vl.x().fieldQ('example_field').bin(true),
  vl.y().count()
)
```

To produce Vega-Lite JSON like this:

```
{
  "mark": "bar",
```

```
 "data": {"url": "data/example.Json"},
 "encoding": {
  "x": {
   "bin": true,
   "field": "example_field",
   "type": "quantitative"
  },
  "y": {
   "aggregate": "count",
   "type": "quantitative"
  }
 }
}
```

5. MySQL: The architectural overview is the SQL Databases component. SQL, or Structured Query Language, is the standard language for relational database management. The databases store all pertinent data, such as user information, educational content, and interaction logs. Through Django's Object-Relational Mapping (ORM) layer, the application interacts seamlessly with the databases, abstracting complex queries into Python code. This integration allows for a robust and secure data storage solution, ensuring data integrity and providing the foundation for the adaptive learning features of the tool.

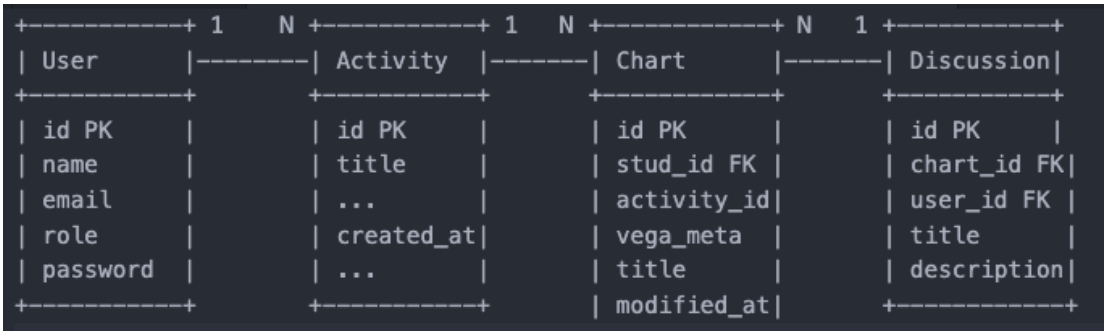


Figure 3.1.2 SQL Data Structure

6. Model-View-Controller: The Model-View-Controller (MVC) [20] architecture is a widely used design pattern for developing user interfaces that divides an application into three interconnected components. This separation of concerns allows for efficient code reuse and parallel development. Below is an overview of each component within the MVC architecture:
    a. Model: The Model is the principal component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic, and rules of the application. Any representation of data such as a database — containing raw data structures or objects — is part of the Model. It responds to the instructions from the controller on the data to present and to the queries from the View about how data should be displayed. It represents the application's response to the request to provide the state of the application.
    b. View: The View represents the UI components. It renders the Model into a form suitable for interaction, typically a user interface element. The View displays data from the Model to the user and enables them to modify it. Multiple Views can exist for a single Model for different user interfaces or

interaction modes. It observes the Model and gets updated when the Model changes, ensuring the presentation layer is separated from the business logic.

c. Controller: The Controller acts as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model, and interact with the Views to render the final output. It listens to the user input (from the View) and performs interactions on the data model objects. The Controller receives the input, validates it, and then performs the business operation that modifies the state of the data model.

The MVC pattern is particularly well-suited for web applications, where the View is the HTML or XHTML output rendered by the browser, the Controller is the part of the system that handles the user's requests (often via HTTP requests), and the Model is the part of the system that handles the logic for the application's data domain. By separating these concerns, developers can work on the Model without affecting the View, and vice versa, leading to a more modular, manageable, and testable application design.

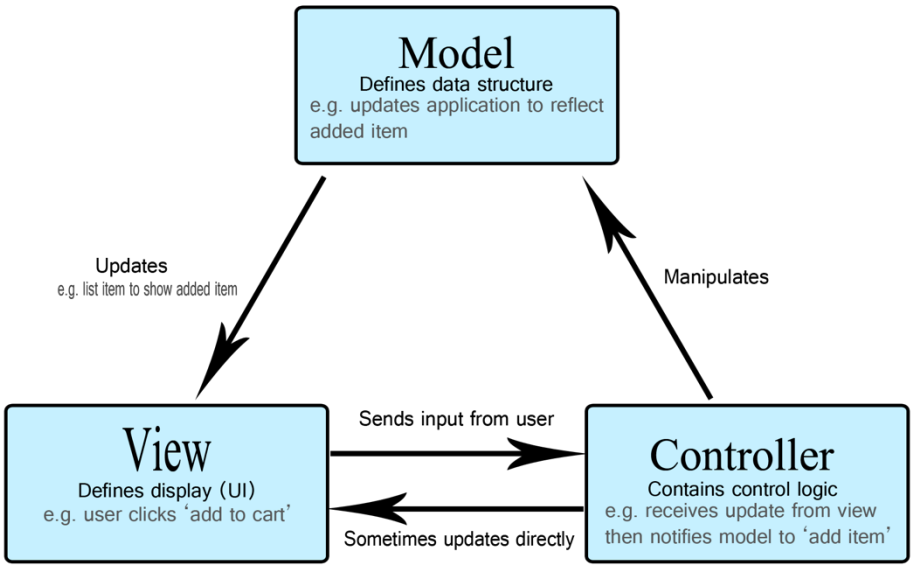


Figure 3.1.3 MVC Design Structure. Source: MDN web Docs[20]

In summary, the backend architecture of the assistive educational tool for data visualization is a composite of well-defined components that work cohesively to offer a seamless educational experience. This constructive interaction between user interfaces, application logic, server infrastructure, and database systems are architected with a focus on scalability, security, and user-centric design, aligning with the pedagogical objectives of the tool.

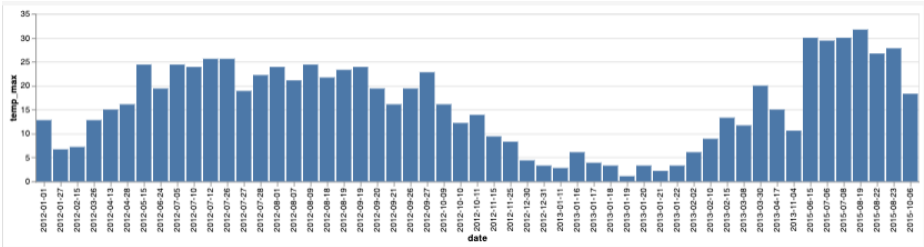
3.2 Frontend

The front end of the system is engineered using AngularJS [21], a structured and dynamic JavaScript framework designed for building rich single-page applications. AngularJS excels in creating scalable and maintainable front-end architectures, making it an apt choice for developing the user interface of the educational tool. Its data binding and

dependency injection eliminates much of the code one would otherwise have to write, thus streamlining the development process.

AngularJS operates with the Django backend via a RESTful API in this architecture, AngularJS clear contract for communication between the front and backend, allowing data to be transferred seamlessly stateless using HTTP requests. The REST API endpoints expose the functionalities of the Django application, such as user registration, activity management, and chart rendering, which the AngularJS frontend consumes.

The integration of AngularJS with the Django REST framework is a powerful combination that enhances the overall user experience. It enables the frontend to request and receive data asynchronously from the backend, updating the web interface dynamically without the need for page reloads. This asynchronicity, powered by AngularJS's built-in services, allows for creating interactive and responsive web applications. Users can engage with the educational tool's features, such as initiating tasks, participating in discussions, and viewing data visualizations in a fluid and intuitive manner.



Id	Date	Precipitation	Temp_max	Temp_min	Wind	Weather
0	2012-01-01	0	12.8	5	4.7	drizzle
26	2012-01-27	0	6.7	-2.2	1.4	drizzle
45	2012-02-15	0	7.2	0.6	1.8	drizzle
85	2012-03-26	0	12.8	6.1	4.3	drizzle
103	2012-04-13	0	15	3.9	4	drizzle
118	2012-04-28	0	16.1	8.3	2.5	drizzle
135	2012-05-15	0	24.4	9.4	4.1	drizzle
175	2012-06-24	0	19.4	9.4	2	drizzle
186	2012-07-05	0	24.4	10.6	3.1	drizzle
191	2012-07-10	0	23.9	11.1	2.3	drizzle

Figure 3.2.1 Frontend Chart with table

Additionally, AngularJS's modular nature facilitates the division of the frontend into reusable components, directives, and services, promoting code reusability and ease of testing. The MVC (Model-View-Controller) architecture inherent in AngularJS ensures that the user interface (View) is decoupled from the application logic (Controller) and data models (Model), resulting in a clean separation of concerns. Given the complexity of data visualization tasks, the frontend is designed to be intuitive and user-friendly. AngularJS assists in implementing dynamic views that react to user interactions and data changes without requiring page reloads. This dynamic updating is essential for educational tools, where immediate feedback and interactive learning environments are pivotal for student engagement and understanding.

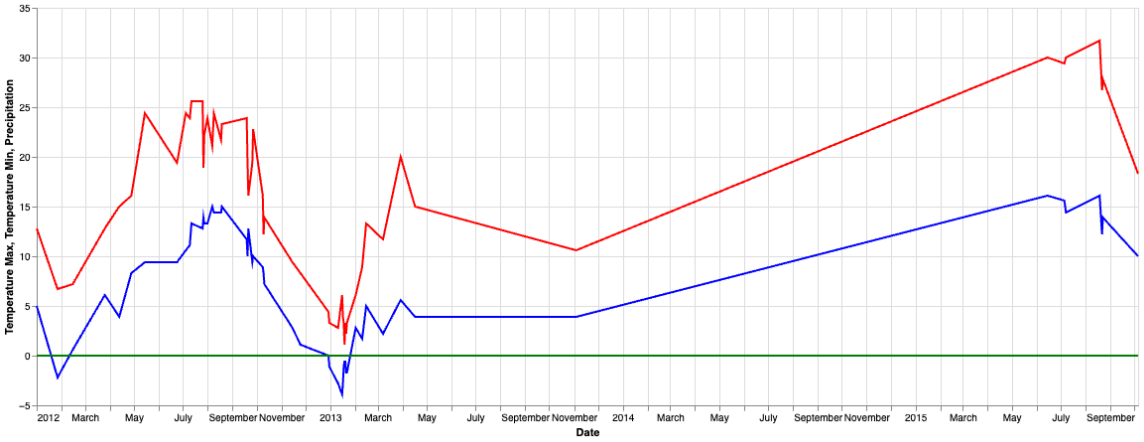


Figure 3.2.2 Multiple Line Chart

In summary, the utilization of AngularJS for the front-end development, in conjunction with the Django framework via a REST API, forms a potent combination that underpins the effectiveness of the assistive educational tool. This constructive collaboration facilitates a powerful and interactive learning experience, with AngularJS enhancing the presentation and interactivity at the frontend, while Django efficiently handles the business logic and data persistence at the backend.

CHAPTER 4

Use Case for the Tool

4.1 Student

The tool is innovatively engineered to enhance students' educational journey in the realm of data visualization, empowering them to unlock actionable insights from raw data. As students upload their dataset, the tool leverages the Vega-Lite API to automatically generate visual representations based on selected chart types and data attributes. This not only visualizes the data but also imparts a practical understanding of how each attribute influences the story the data tells.

Incorporating an ngstack code editor within the Angular framework, the tool provides an interactive coding environment. This environment is essential for demystifying the Vega-Lite JSON syntax and helping students appreciate the nuances of data-driven storytelling. By tweaking the JSON code, they gain insights into the impact of their coding decisions on the visual output, thereby fostering a deeper comprehension of the visualization process.

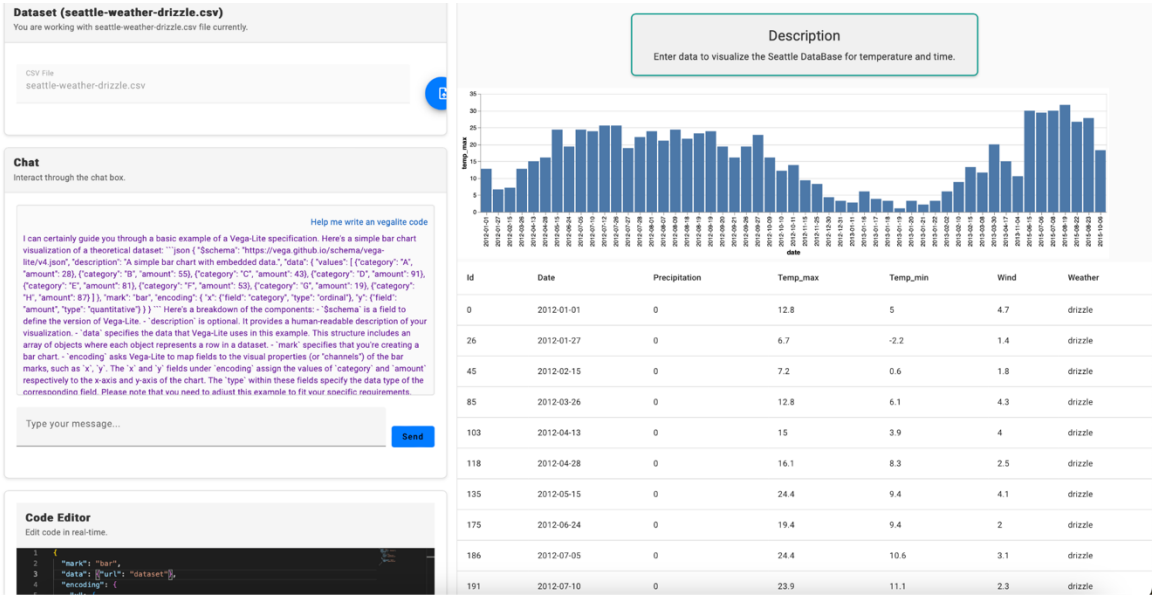


Figure 4.1.1 Tool Interface

Beyond mere visualization, the tool encompasses an Integrated Large Language Model (LLM) Chatbot that serves as an on-demand tutor. This chatbot engages students in a dialogue about visualization strategies, guiding them through the logical steps to represent data effectively. It also critiques their current JSON structures, offering constructive suggestions for enhancement. This personalized feedback loop is crucial for students to iterate on their designs, encouraging a mindset of continuous improvement.

The chatbot is designed to understand the students' queries contextually, allowing for a more nuanced and focused guidance. For example, if a student struggles with selecting the right type of chart for their data, the chatbot can provide a walkthrough of the best practices in chart selection based on the data's characteristics and the insights sought.

The diagram represents the entities involved in the system and their relationships:

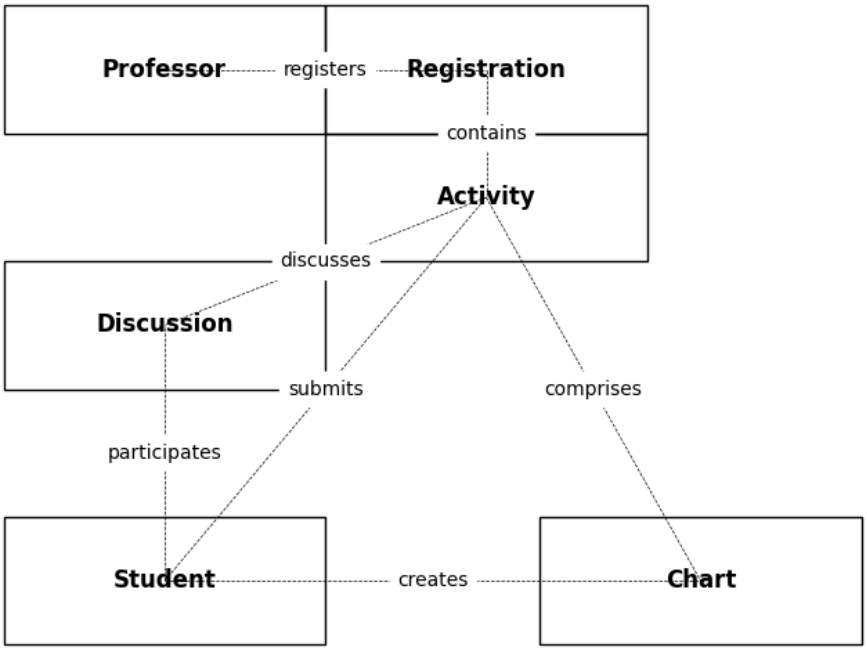


Figure 4.1.2 Entity Relational Diagram

- Professor: Registers for the system.
- Registration: Contains multiple activities.
- Activity: Is comprised of charts and is discussed in discussions.
- Student: Participates in discussions, submits activities, and creates charts.
- Discussion: Is a platform for students to engage in discussions about activities.
- Chart: Is an element created by students as part of their activity submissions.

4.2 Professor

In this educational ecosystem, the professor is empowered with administrative capabilities to facilitate a seamless learning environment. Tasked with the critical role of activity registration, the educational content by linking pertinent charts and fostering discussions tailored to the users' input and interactions. As students engage with the interactive components—partaking in dialogues, exercises, and visual analysis—the admin portal provides the professor with a comprehensive oversight of these engagements.

The administrative interface is adept at capturing and displaying the Vega-Lite JSON strings that students can interact with and is current JSON structure is stored in the MySQL database. This functionality is pivotal for professors to monitor and assess students' progress in real-time. By having immediate access to the visualization code and its resulting output, the professor can deliver instantaneous, targeted feedback, offering constructive insights to enhance the students' learning trajectory.

Additionally, this administrative vantage point allows the professor to gauge the effectiveness of teaching methodologies, adapt to the collective progress of the class, and individualize the learning experience where necessary. This real-time feedback loop is instrumental in creating a dynamic classroom atmosphere that is responsive to the students' educational needs, ultimately leading to a more robust understanding of data visualization principles and practices.

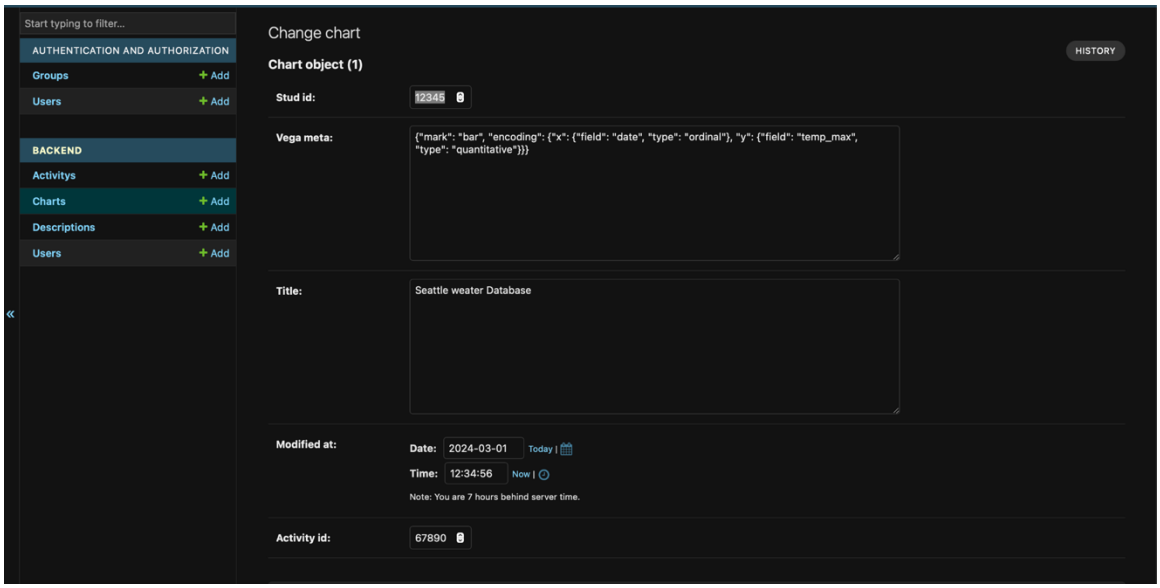


Figure 4.2.1 Django Admin View for the Chart model

CHAPTER 5

Conclusions and Future Work

5.1 Conclusion

This thesis was carefully planned with the main goal of improving the pedagogy related to data visualization to contribute to educational technology. It aimed to answer this need by developing an inventive assistive educational tool. It recognized the urgent need for efficient teaching tools that can adjust to the changing requirements of a varied student body, including those with impairments.

The thesis conducted a detailed analysis of the most recent technological developments in data visualization tools as well as the instructional techniques that are now in use. It put forth a paradigm that makes use of these findings to provide inclusive, productive, and user-centric learning experiences. The creation and execution of a tool that combines the strength of contemporary online technology with tried-and-true instructional methodologies represents the pinnacle of this effort.

The thesis makes a substantial contribution by showing how a well-designed software system that adheres to the Model-View-Controller (MVC) design pattern can make it easier to create an interactive, scalable, and maintainable tool. A Django-based backend and AngularJS front end integrated through a RESTful API demonstrated the easy blending of technologies to produce a responsive and interesting learning environment.

The thesis also explored the complexities involved in developing a flexible learning environment that honors each learner's uniqueness. Through the integration of adaptive

learning concepts, the tool offers customized learning experiences by customizing the content delivery and learning pace to meet the individual needs of every learner. This method completely changes the way that data visualization is explained and taught.

The teaching aid created for this thesis not only helps students learn the art and science of data visualization, but it also gives teachers the tools they need to impart knowledge more skillfully. Technology can improve learning outcomes because of its capacity to provide real-time feedback, support a variety of learning paths, and promote active engagement.

In conclusion, the thesis has successfully shown that the thoughtful application of educational theories, user-centered design, and technological innovation can profoundly enhance the educational landscape of data visualization. It has laid the groundwork for future research and development in the domain, opening avenues for further enhancements and validation through empirical studies. As the field of data visualization continues to grow in importance across academic and professional disciplines, the work completed in this thesis will undoubtedly serve as a valuable resource for those seeking to advance the efficacy of data visualization education.

5.2 Future Works

Building upon the foundation laid by this thesis, there are several promising directions for future research and development in the field of educational tools for data visualization. The exploration and enhancement of the assistive tool developed in this thesis can continue along various avenues:

1. Empirical Validation: Carrying out extensive user research and trial experiments in the classroom to evaluate the tool's efficacy in several learning environments. utilizing statistical analysis and actual data to measure the tool's effect on learning outcomes and student engagement.
2. Open-Source Community Development: Converting the project to an open-source framework so that global educators and developers can participate in the tool's creation. Creating a community around the tool to exchange improvements, bespoke modules, and best practices.
3. Integration with Other Educational Tools: Ensuring that Learning Management Systems (LMS) and other educational platforms used by universities worldwide may be readily integrated with it. creating extensions and plugins that facilitate a smooth interaction across various educational software ecosystem
4. Integration of LLMs: Utilizing large language models to give the instructional tool's assistance prompts intelligence and context. creating an interface that enables these models to interpret user input and provide real-time support and direction for learning. Using LLMs' sophisticated natural language processing powers to analyze student input and offer clarifications, recommendations, or criticism to promote a deeper comprehension of data visualization principles. putting in place a feedback loop so that the LLM can gradually increase the usefulness and relevance of its prompts by learning from student interactions. Integrating LLMs with educational objectives and promoting positive learning outcomes is ensured by investigating ethical issues and establishing ethical AI procedures. Using large language models is an innovative approach to improving the adaptability and interactivity of

educational tools. By incorporating these sophisticated AI elements, the tool can become even more responsive to individual learner needs, providing a highly personalized and supportive educational environment.

REFERENCES

1. Mayer, R. E. (2009). "Multimedia Learning." Cambridge University Press.

2. "Using Data Visualization to Find Insights in Data." DataJournalism.com, [n.d.], datajournalism.com/read/handbook/one/understanding-data/using-data-visualization-to-find-insights-in-data. Accessed [date you accessed the site]

3. Adaptive Learning." Montclair State University, Montclair State University, n.d., www.montclair.edu/itds/digital-pedagogy/pedagogical-strategies-and-practices/adaptive-learning/.

4. Dewey, J. (1938). Experience and Education. New York: Kappa Delta Pi.

5. Vygotsky, L.S. (1978). Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press

6. Wilkinson, L. (2005). The Grammar of Graphics (2nd ed.). Springer.

7. "Tableau Software." Tableau Software, n.d., www.tableau.com/.

8. "Power BI." Microsoft, n.d., powerbi.microsoft.com/

9. "Google Data Studio." Google, n.d., datastudio.google.com/.

10. "ggplot2: Elegant Graphics for Data Analysis." ggplot2.tidyverse.org, n.d., ggplot2.tidyverse.org/.

11. Matplotlib: Python plotting." Matplotlib, n.d., matplotlib.org/. Accessed [Your Access Date].

12. "seaborn: statistical data visualization." Seaborn, n.d., seaborn.pydata.org/

13. "D3.js - Data-Driven Documents." D3js, n.d., d3js.org/.

14. "Gephi - The Open Graph Viz Platform." Gephi, n.d., gephi.org/.

15. "Qlik Sense." Qlik, n.d., www.qlik.com/us/products/qlik-sense.

16. "Datawrapper: Create charts, maps, and tables." Datawrapper, n.d., www.datawrapper.de/.

17. "Infogram: Create Infographics, Reports and Maps." Infogram, n.d., infogram.com/.

18. "Django." Django Software Foundation, n.d., www.djangoproject.com/.

19. "Vega-Lite: A Grammar of Interactive Graphics." Vega-Lite, edited by the Vega-Lite Team, n.d., vega.github.io/vega-lite/.

20. "MVC." MDN Web Docs, Mozilla, n.d., developer.mozilla.org/en-US/docs/Glossary/MVC.

21. "AngularJS." AngularJS, n.d., angularjs.org