**Admission Compass**

**Table of Contents**

1. Introduction
2. Goals and Objectives
3. Functionality Requirements
4. Non-Functionality Requirements
5. Detailed Design
6. Literature Review
7. Competitive Analysis
8. Metrics
9. Algorithms
10. User Interface Design
11. Database Design
12. Use Cases
13. Conclusion

**1. Introduction**

Starting the journey of higher education can be intimidating and overwhelming. Students face challenges in making decisions about career paths, finding the right schools, and navigating through diverse admission processes. Our project, Admission Compass, aims to support students by providing clear, accessible information about admission requirements, empowering them with knowledge and easing their journey into higher education.

**2. Goals and Objectives**

**Goals:**

* Offer support in the admission process.
* Ensure access to information about available paths to admission.
* Empower students with knowledge about available admission requirements.

**Objectives:**

* Develop a user-friendly website to help users explore degree programs and their admission criteria.
* Create a platform for former students to share their admission criteria.
* Implement a search tool for users to refine searches by degree, institution, and personal scores.
* Utilize built-in tools, functions, and machine learning to present and predict data.

**3. Functionality Requirements**

**Admission Compass Features:**

1. **Previous Student Admissions Form & Table:** Collect and store important information from previously admitted students, including institution, degree type, Bagruyot subjects, scores, psychometric test scores, and additional courses.
2. **Institutions & Institutions’ Degrees Tables:** Store essential data about institutions and degrees, such as program name, type, admission criteria, and general information.
3. **Search Tool:** Enable users to explore admission requirements based on degree, institution, or personal grades using predefined dropdown menus for accuracy.
4. **List of Search Results:** Display search results in a structured list format for easy scanning.
5. **Detailed Result Pages:** Offer comprehensive information about admission requirements, including institutional data, degree specifics, and insights from past students.

**4. Non-Functionality Requirements**

**User Experience:**

* Easy-to-use interface with straightforward navigation and clear instructions.
* Minimalist yet visually pleasing design to avoid overwhelming users.
* Information displayed concisely or in detail, depending on user needs.

**Performance:**

* Quick responses to user queries, ensuring prompt delivery of search results and interactions.

**Scalability:**

* Our system doesn't encounter bottlenecks in user capacity as it efficiently handles incoming user data via a specialized interface. Moreover, it exhibits scalability, enhancing its performance with an influx of new users. Notably, our models continually improve in accuracy as they aggregate more data over time.

**Maintainability:**

* Clean, organized, and well-commented code.
* Continuous testing and monitoring for high performance and reliability.
* Efficient database management, regular updates, and new feature additions.

**5. Detailed Design**

**Physical Architecture:**

* **User Interface:** Users access the website via web browsers on desktops, laptops, tablets, and smartphones.
* **Web Server:** Hosts the frontend code (HTML, CSS, JavaScript, React, TypeScript).
* **Application Server:** Hosts backend code, processes data, interacts with the database and machine learning server.
* **Machine Learning Model Server:** Hosts and serves machine learning models, handles prediction requests.
* **Database Server:** Stores and manages website data, including student data, school information, and degree details.
* **Admin:** Manages website content, database tasks, and structural modifications.

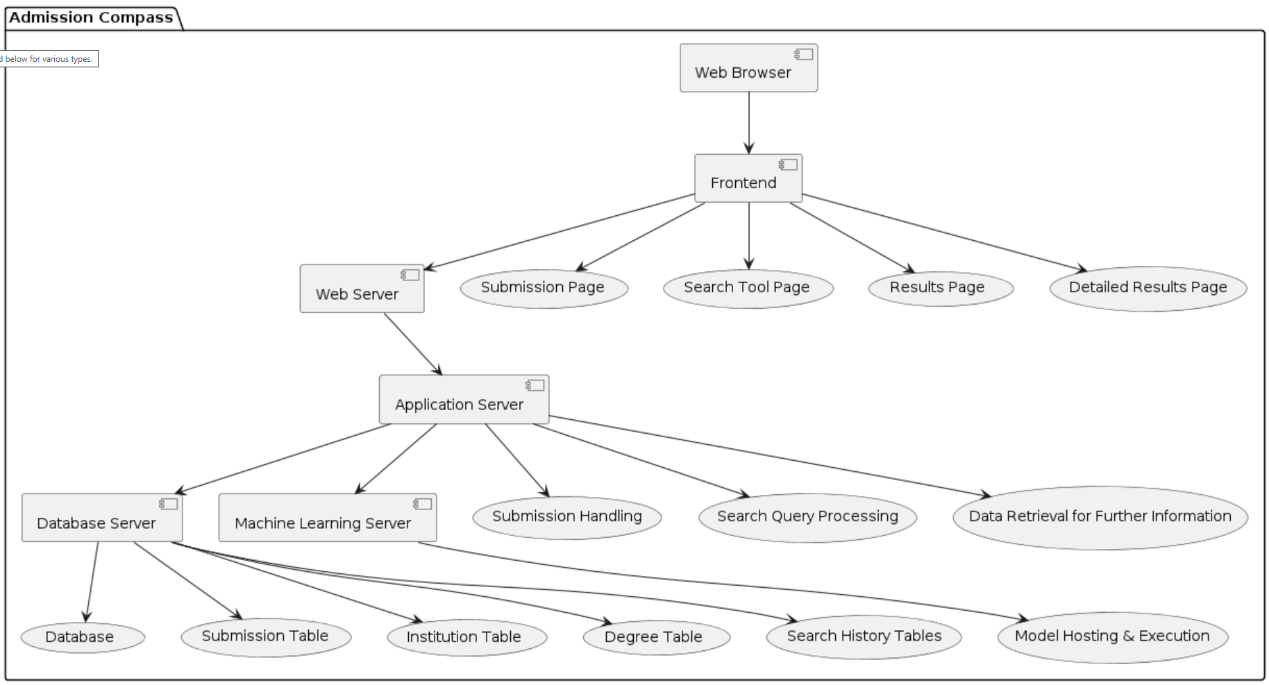
**Component Diagram Overview:** Here we have the various components and interactions within our system. It provides a high-level view of the system architecture, highlighting how different parts of the system communicate and work together.

A diagram of a diagram

Description automatically generated

**Components:**

1. **Web Browser:**
   * **Description:** The user interface through which users access the Admission Compass website.
   * **Interaction:** Interacts with the Frontend component.
2. **Frontend:**
   * **Description:** The user-facing part of the system that includes pages like the Submission Page, Search Tool Page, and the Detailed Results Page.
   * **Interaction:** Hosted on the Web Server and it communicates with the Application Server using Node.js.
3. **Web Server:**
   * **Description:** Hosts the frontend code (HTML, CSS, TypeScript, React) and displays it to the user's web browser. Handles the logic connecting to the backend, processes data, and interacts with the database server, and the Machine Learning Server.
   * **Interaction:** Sends requests from the users to the Database Server and machine learning server and receives data back and displays it on the website.
     + **Submission Handling:** Processes and sends user submissions to the database.
     + **Search Query Processing:** Executes search queries and retrieves Acceptance Classification Predictions and returns Shap Plot Explainability Visuals, as well as data analysis over stored data.
4. **Database Server:**
   * **Description:** Manages data storage and retrieval for the system.
   * **Sub-components:**
     + **Institution Table:** Contains institution information.
     + **Degree Table:** Stores degree program details.
   * **Interaction:** Communicates with the Application Server.
5. **Machine Learning Server:**
   * **Description:** Hosts and executes machine learning models that provide predictive analytics for the system.
   * **Sub-components:**
     + **Model Hosting & Execution:** Manages the ML models and processes prediction requests.
   * **Interaction:** Communicates with the Application Server.



**6. Literature Review**

**Background:** The journey of higher education can be daunting. High admission standards and a multitude of choices add to the complexity. However, universities are broadening their admission criteria to be more inclusive. Despite these changes, many students are unaware of the alternative paths available to them.

**Key Insights:**

* **Higher Education Trends:** Enrollment is rising due to cultural values and institutional marketing efforts. More students apply to multiple institutions, increasing costs and competition.
* **Standardized Tests:** Traditional reliance on standardized tests is being challenged. Combining test scores with high school performance and personal attributes offers better predictions of academic success.
* **Alternative Admission Criteria:** Institutions are considering a variety of factors, including high school course rigor, character, and intellectual curiosity, to create diverse and successful student bodies.
* **7. Competitive Analysis**

**Competitors:**

1. **לימודים בישראל (Studies in Israel):**
   * **Strengths:** Comprehensive information, prediction tool for admission likelihood.
   * **Weaknesses:** Limited data inputs, confusing layout.
2. **College Scoreboard:**
   * **Strengths:** Clear website, streamlined search process, visual data comparison.
   * **Weaknesses:** Limited personal data inputs, no predictive tools.
3. **CollegeData.Com:**
   * **Strengths:** Detailed information, robust admission calculator.
   * **Weaknesses:** Overwhelming amount of information, unclear visualization, requires account creation.

**Website Strategy:**

* Focus on admission requirements for Israeli institutions.
* Collect thorough admission data from past and current students.
* Implement a user-friendly search tool with filter capabilities.
* Integrate an admissions prediction tool with multiple criteria inputs.

**8. Metrics**

**Accepted Student Submissions:** Track the number of submissions to gauge user engagement.

**User Feedback:** Gather feedback through surveys and reviews to assess user satisfaction and website success.

**9. Algorithms**

**Predicting Acceptance Algorithm:**

* Utilizes machine learning techniques to predict acceptance probability based on student’s grades.
* Processes submission data, cleans and transforms it, trains models, and predicts acceptance probabilities.

**Predicting Acceptance Algorithm:**

* **Description:** This algorithm predicts the probability of acceptance into a program based on a student's grades.
* **Models Used:**
  + **Decision Trees:** A tree-like model of decisions and their possible consequences.
  + **Random-**Forest:
  + **Gaussian:**
* **Process:** The algorithm uses submission data from past applicants, preprocesses the data, trains models, and predicts acceptance probabilities. It calculates a weighted average from the predictions to generate an overall prediction score. The weights are determined by the accuracy levels of each model.

**Academic Feature Importance Algorithm:**

* **Description:** Identifies the most influential academic features in the admissions process for a school or degree program.
* **Techniques Used:**
  + **SHAP (SHapley Additive exPlanations) Values:** A method to explain the output of machine learning models.
  + **Voting Algorithm:** Combines the SHAP results computed using the different models and “votes” on the elected features.
* **Process:** Utilizes trained models to analyze feature importances, helping prospective students understand which requirements are crucial for acceptance.

**Statistics Generators-**

**Calculating Averages and Ranges:**

* **Description:** Computes averages and ranges of academic scores for institutions, degrees, and degree types.
* **Techniques Used:**
  + **Aggregation:** Sums up scores and divides by the total number of submissions to calculate averages.
  + **Range Calculation:** Identifies minimum and maximum values within each criterion.
* **Process:** Aggregates and analyzes scores to provide users with clear insights into academic requirements.

**Aggregating Students Acceptance Criteria:**

* **Description:** Showcases the statistics of academic requirements that lead to acceptance by institutions and specific degrees.
* **Techniques Used:**
  + **Combination Identification:** Identifies all possible combinations of acceptance criteria.
  + **Occurrence Tallying:** Counts occurrences of each combination among accepted submissions.
* **Process:** Presents the results in bar charts, illustrating the distribution of acceptance criteria to each school/degree.

**10. User Interface Design**

**Student Upload Page:** Allows users to submit their academic details.

**Search Tool Page:** Enables users to search for admission requirements based on various criteria- by institution, by degree, or by grades. Results appear below the search tool.

**Detailed Results Page:** Provides comprehensive details about admission requirements and acceptance probabilities for the specified result.

**11. Database Design**

Database Schema: The following tables outline the structure of our database, including the primary keys (PKs) and foreign keys (FKs) for each table.

* **institution:** Contains information about institutions, such as name, location, and type.
* **degrees:** Stores details about degrees offered by institutions, including name, and type.
* **bagruyot:** Contains Bagruyot (high school matriculation exam) scores and units for various subjects per student.
* **psychometric\_scores:** Stores psychometric exam scores, including general grade, verbal score, English score, and quantitative score per student.
* **student\_submissions:** Holds submissions from accepted students, including their email, degree choices, institution acceptance/rejection, Bagruyot ID, psychometric scores, and other academic details.
* **acadmic\_courses:** Contains details about college/academic courses that students may have taken.
* **preparatory\_courses:** Contains details about courses within preparatory programs, including name and description.
* **courses:** Stores general course information, including name and description.
* **academiccourses:** Contains information about academic courses, including their names.

**Database Schema:** The following tables outline the structure of our database, including the primary keys (PKs) and foreign keys (FKs) for each table.

**Entity-Relationship Diagram (ERD):** The ERD provides a visual representation of the database schema. It illustrates how tables such as **institution**, **degrees**, **bagruyot**, **psychometric\_scores**, **student\_submissions**, **preparatory\_courses**, and **academiccourses** are related through primary keys (PKs) and foreign keys (FKs).

**12. Use Cases**

**Use Case 1: Exploring Admission Requirements**

* **Actor:** Prospective Student
* **Goal:** To explore admission requirements for universities and degree programs in Israel.
* **Steps:**
  1. The prospective student searches for admission requirements using the Admission Compass website.
  2. The system presents a list of search results based on the input criteria.
  3. The student selects a specific program from the search results.
  4. The system provides detailed information about the selected program.
  5. The student can refine their search using various filters to narrow down the results.

**Use Case 2: Previously Accepted Student Submission**

* **Actor:** Previously Accepted Student
* **Goal:** To submit admission criteria and other relevant information to the Admission Compass website.
* **Steps:**
  1. The accepted student fills out a submission form with their admission criteria and other details.
  2. The system validates the entered information to ensure completeness and accuracy.
  3. The validated information is stored in the database.
  4. The system displays a confirmation message to the student indicating successful submission.

**Use Case 3: Data Analysis and Website Improvement**

* **Actor:** System Administrator
* **Goal:** To analyze submitted data and use insights to enhance the Admission Compass website.
* **Steps:**
  1. The system administrator retrieves submitted data from the database.
  2. The system analyzes the data to identify trends and patterns.
  3. The system identifies areas for potential improvement based on the analysis.
  4. The identified improvements are implemented on the website.
  5. The effectiveness of the implemented improvements is verified through testing and monitoring.

A diagram of a process

Description automatically generated

**13. Conclusion**

Admission Compass aims to support students by providing clear, accessible information about admission requirements. By leveraging advanced search tools, predictive algorithms, and user-friendly design, we strive to empower students with the knowledge they need to navigate their higher education journey confidently.

A diagram of a diagram

Description automatically generated