International Journal of Cyber and IT Service Management (IJCITSM)
Vol. 6, No. 2, October 2026, pp. 172−184
E-ISSN: 2808-554X | P-ISSN: 2797-1325, DOI:10.34306

A Modular Business Intelligence Architecture for Predictive Student Admission Decision Support

Mohamad Rakhmansyah[1], Untung Rahardja[2], Muhammad Faizal Fazri[3*], Nikita Ivanov[4]
[1,3]Faculty of Science and Technology, Universitas Raharja, Indonesia
[2]Faculty of Economics and Business, Universitas Raharja, Indonesia
[4]Pandawan Group, New Zealand
[1]rakhmansyah@raharja.info, [2]untung@raharja.info, [3]faizal.fazri@raharja.info, [4]inva.niki@pandawan.ac.nz
***Corresponding Author**

Article Info

Article history:

Submission June 3, 2026
Revised July 13, 2026
Accepted September 14, 2026
Published September 30, 2026

Keywords:

Artificial Intelligence
Business Intelligence
Predictive Analytics
System Architecture
Student Admission

ABSTRACT

Student admission management in higher education increasingly depends on data-driven. Systems that can support predictive decision making beyond descriptive dashboard reporting. Existing Business Intelligence dashboards commonly display admission indicators such as leads, applicants, registrations, targets, program distribution, and historical trends, but they often do not include predictive processing, risk classification, warning generation, or recommendation outputs. **This study aims** to design an Artificial Intelligence (AI) Business Intelligence architecture for predictive student admission decision support. **Using a design** oriented model development approach, the proposed architecture integrates admission data collection, preprocessing, Prophet based forecasting, risk scoring, Representational State Transfer (REST) API communication, and dashboard wireframe design. The system architecture consists of Vue.js as the presentation layer, Laravel as the application and API layer, MySQL as the data layer, and Python as the predictive analytics layer. Admission records are transformed into forecasting-ready datasets., processed through Prophet time series prediction, classified into program level risk categories, and converted into warning and recommendation outputs. **The proposed model** contributes a technical data pipeline and modular predictive workflow that can extend existing descriptive admission dashboards without replacing the entire information system. **The architecture aligns** with SDG 4 on Quality Education by supporting informed admission planning and with SDG 9 on Industry, Innovation and Infrastructure by promoting modular digital innovation in higher education. Future research should implement the architecture using longitudinal admission datasets, compare Prophet with Autoregressive Integrated Moving Average (ARIMA), Random Forest, and XGBoost, evaluate forecasting accuracy, test API performance, and assess dashboard usability in operational admission management.

This is an open access article under the CC BY 4.0 license.

DOI: https://doi.org/10.34306/ijcitsm.v6i2.284
This is an open-access article under the CC-BY license (https://creativecommons.org/licenses/by/4.0/)
©AAuthors retain all copyrights.

1. INTRODUCTION

Business Intelligence dashboards are commonly used to support student admission monitoring [1]. Through dashboards, admission managers can visualize the number of leads, applicants, registered students, target achievement, program level distribution, and previous period comparison [2, 3]. These dashboards im-

prove data visibility and make admission information easier to access [4]. However, most admission dashboards remain descriptive because they mainly present current and historical indicators. They show what has happened, but they do not process the data into predictive outputs that can estimate what is likely to happen next [5].

Predictive decision support requires a more technical data workflow. Admission data must be collected, cleaned, aggregated, transformed into forecasting-ready datasets., and processed using a predictive model [6, 7]. The system also needs to classify risk, generate warning signals, and produce recommendation outputs that can be interpreted by users [8, 9]. Without this workflow, dashboard users still need to interpret admission patterns manually. For example, a dashboard may show low registration in a study program, but it does not automatically estimate the final registration outcome, classify the program's risk level, or suggest whether campaign intensification or applicant follow-up is required [10].

Artificial Intelligence and predictive analytics can extend Business Intelligence dashboards by adding forecasting and decision support functions. In this study, predictive analytics is positioned as a technical service connected to the existing dashboard architecture. Vue.js is used as the presentation layer, Laravel as the application and Representational State Transfer (REST) API layer, MySQL as the data layer, and Python as the predictive analytics layer. Prophet based time series forecasting is proposed to estimate registration trends, while risk scoring, threshold based warning, and rule based recommendation logic are used to transform prediction results into operational decision support outputs [11].

Previous studies have discussed BI dashboards and AI based decision support in higher education [12, 13]. However, limited studies have described a technical architecture that connects admission data collection, preprocessing, forecasting, risk scoring, API based prediction service, and dashboard presentation in one integrated system [14, 15]. This gap is important because predictive admission support requires not only an attractive dashboard interface but also a clear data pipeline and modular system architecture capable of transforming admission records into forecasting and risk outputs [16].

Beyond its technical contribution, the proposed architecture is relevant to the Sustainable Development Goals. By supporting earlier identification of admission shortfalls, evidence based recruitment interventions, and more informed institutional capacity planning, predictive admission decision support can contribute to SDG 4 on Quality Education [17]. The modular integration of existing information systems with AI based analytics, Representational State Transfer (REST) API services, and reusable dashboard components also supports SDG 9 on Industry, Innovation and Infrastructure by strengthening digital infrastructure and institutional innovation in higher education. These contributions remain prospective and require validation through operational implementation and evaluation.

Based on this gap, this study aims to design an Artificial Intelligence (AI) Business Intelligence architecture for predictive student admission decision support [18]. Unlike previous BI dashboard studies that mainly visualize current and historical admission indicators, and AI based decision support studies that frequently examine predictive models as isolated analytical components, this study proposes an integrated end-to-end architecture.that connects admission data collection, preprocessing, Prophet based forecasting, program level risk classification, threshold based warning generation, rule based recommendations, Representational State Transfer (REST) API communication, and reusable dashboard components [19]. The novelty of the study lies in its modular and incrementally deployable architecture, in which an independent Python predictive service is integrated with an existing Vue.js, Laravel, and MySQL admission system without requiring replacement of the institution's core information system [20]. The study therefore contributes an implementation oriented architectural blueprint for transforming descriptive admission dashboards into predictive decision support systems.

2. METHODOLOGY

This study employed a design oriented model development approach to formulate a technical architecture and predictive workflow for student admission decision support [21]. This approach was selected because the study focuses on designing a system model rather than testing statistical relationships or evaluating governance outcomes [22]. The design process includes the identification of admission data sources, preprocessing requirements, predictive model components, system architecture, and dashboard wireframe structure. The result of this method is a conceptual architecture that can be used as a technical reference for developing an Artificial Intelligence (AI) Business Intelligence system in student admission management [23].

2.1. Data Collection and Input Sources

The proposed architecture uses admission related datasets obtained from the existing admission information system and dashboard [24]. These datasets represent the flow of prospective students from initial lead acquisition to final registration. The main data sources include lead data, applicant data, registration data, target data, program data, and historical trend data [25]. Each data source has a specific function in the predictive workflow. Lead and applicant data support acquisition and conversion monitoring, registration data become the main forecasting input, target data support achievement analysis, program data enable program level classification, and historical trend data support model training and comparison.

Table 1. Admission Data Sources and Prediction Functions

Data Source	Main Variables	Data Type and Frequency	Prediction Function
Lead data	Date, source channel, program interest	Event level, categorical, and temporal; aggregated daily or weekly	Supports acquisition trend analysis
Applicant data	Application date, program choice, status	Event level, categorical, and temporal; aggregated daily or weekly	Supports applicant to registration monitoring
Registration data	Registration date, program, status	Event level and numerical; aggregated daily or weekly	Provides the main input for forecasting
Target data	Annual target, program target	Numerical; defined annually or for each admission cycle	Supports predicted target achievement and gap analysis
Program data	Program name, faculty, quota	Categorical and numerical; updated for each admission cycle	Supports program level risk classification
Historical trend data	Previous period registration	Numerical time series data; aggregated daily, weekly, or monthly	Supports trend comparison, model training, and validation

Table 1 presents the admission data sources and their corresponding prediction functions in the proposed Artificial Intelligence (AI) Business Intelligence architecture. The table shows that each type of admission data has a specific role in supporting predictive analysis and decision making. Lead data contains information such as date, source channel, and program interest, which is used to analyze acquisition trends. Applicant data includes application date, program choice, and status, supporting the prediction of applicant to registration conversion. Registration data records registration date, program, and status, and is used to estimate future registration volumes. Target data provides annual targets by program, which enables comparison between predicted admissions and institutional targets. Finally, program data contains information about the program name, faculty, and quota, supporting program-level risk classification and admission decision-making. Historical trend data, consisting of previous registration periods, provides the time series information required for model training, trend comparison, and validation. Overall, the table demonstrates how heterogeneous admission data are transformed into structured inputs for forecasting, risk assessment, and predictive decision support.

2.2. Data Preprocessing Procedure

Before prediction is performed, raw admission data need to be prepared into a structured analytical dataset. The preprocessing procedure includes removing duplicate records, standardizing admission period formats, grouping registration data by date, week, or month, aggregating registration data by study program, aligning registration data with target data, and handling missing values [26, 27]. These steps are required to ensure that the predictive model receives consistent and reliable input.

For Prophet forecasting, registration records are transformed into a time series structure consisting of date and registration count [28]. The time variable represents the admission period, while the registration count represents the number of confirmed registrations within a specific time interval. Program level datasets are prepared separately by combining registration data, program data, target data, and historical trends. This dataset is used to support risk classification, early warning, and recommendation generation.

2.3. Predictive Model Design

The predictive model consists of forecasting, risk scoring, early warning, and recommendation modules. Prophet is proposed as the initial forecasting method because student registration data can be structured as a time series using the date variable ds and registration count y [29, 30]. Daily or weekly aggregation will be used depending on data availability. The model will estimate registration trends and final target achievement.

The proposed Prophet configuration includes linear growth, weekly seasonality when daily data are available, and institutional events such as campaigns, scholarship announcements, entrance tests, and registration deadlines. Key parameters, including changepoint prior scale and seasonality prior scale, will be tuned during implementation. Validation will use chronological train test splitting and rolling origin evaluation. Forecasting performance will be measured using MAE, RMSE, and MAPE.

Prophet is selected as the initial baseline because it provides interpretable trend, seasonality, and changepoint components [31]. However, it will be compared with Autoregressive Integrated Moving Average (ARIMA), Random Forest, and XGBoost using identical datasets, forecasting horizons, and evaluation metrics.

Program risk will be determined using the predicted achievement ratio:

$$PAR_i = \frac{F_i}{T_i} \times 100\% \tag{1}$$

Where PAR_i is the predicted achievement ratio for study program i, F_i is the predicted final number of registrations, and T_i is the admission target. A program is classified as high risk when $PAR_i < 60\%$, medium risk when $60\% \leq PAR_i < 80\%$, and low risk when $PAR_i \geq 80\%$. The risk category is increased by one level when the recent registration trend is declining and the predicted result is lower than the previous admission period.

For example, if a study program has a target of 100 students and the predicted final registration is 54 students, the predicted achievement ratio is 54%, resulting in a high risk classification. Predicted registrations of 72 and 88 students against the same target produce medium and low risk classifications, respectively. High risk programs trigger campaign intensification, applicant follow-up, and recruitment channel review. Medium risk programs require weekly monitoring and targeted follow-up, while low risk programs maintain the current recruitment strategy with routine monitoring.

2.4. Model Development Procedure

The model development procedure consists of five stages that provide a systematic framework for developing the proposed predictive Business Intelligence architecture. The process begins with data source identification, which defines the required lead, applicant, registration, target, program, and historical trend datasets to represent the student admission process. The second stage involves data preprocessing design, where admission records are cleaned, aggregated, and transformed into structured datasets suitable for predictive analysis. The third stage focuses on prediction model design by defining Prophet forecasting, risk classification, alert generation, and recommendation logic to transform historical admission patterns into predictive insights and actionable information.

The fourth stage is system architecture design, which connects Vue.js, Laravel, MySQL, and Python through Representational State Transfer (REST) API communication. The integration of these technologies enables the presentation, application, data management, and predictive analytics components to communicate within a modular system architecture. The fifth stage is wireframe component design, which defines how prediction outputs are presented through dashboard components such as summary cards, forecasting chart, risk table, warning panel, and recommendation panel. These components are designed to organize predictive information in a clear and accessible format so that admission managers can quickly monitor key indicators, identify potential risks, and obtain recommendations to support data driven admission planning. Overall, the five stage procedure establishes a continuous development flow from raw admission data to predictive outputs and their visualization, providing a structured foundation for implementing the proposed AI-driven Business Intelligence system.

2.5. System Architecture Design

The proposed system architecture consists of four main layers: presentation layer, application layer, data layer, and analytics layer. The presentation layer uses Vue.js to render dashboard wireframe components. This layer is responsible for presenting admission information and predictive outputs in a structured and user-friendly interface, allowing users to monitor admission indicators and interpret the results generated by the system. The application layer uses Laravel to manage routing, authentication, API endpoints, and business logic. This layer coordinates the interaction between users, stored data, and analytical services, ensuring that system processes are managed consistently and securely. The data layer uses MySQL to store admission, registration, target, program, and historical datasets. The centralized storage enables the system to maintain structured admission records and provide historical data required for forecasting and comparative analysis.

The analytics layer uses Python to perform forecasting, risk scoring, alert processing, and recommendation logic. This layer transforms the available admission data into predictive information that can be used to identify potential risks, generate early warnings, and provide recommendations for admission management. Communication between the Laravel application layer and Python analytics layer is handled through Representational State Transfer (REST) API integration. This integration enables the exchange of input data and analytical results between the application and prediction components while maintaining a modular separation between the main system and the predictive processing functions. Through the integration of these layers, the proposed architecture provides a systematic workflow from data storage and processing to predictive analysis and visualization, supporting more informed and data driven student admission decision-making.

Table 2. Proposed Technology Stack, System Functions, and Selection Justification

Layer	Technology	Technical Function	Selection Justification
Presentation layer	Vue.js	Renders predictive dashboard components and user interactions	Selected for its reactive component model, reusable interface structure, and compatibility with the existing frontend; React and Angular are possible alternatives.
Application layer	Laravel	Manages routing, authentication, API endpoints, data aggregation, and business logic	Selected for its mature web framework, built in security features, Representational State Transfer (REST) API support, and compatibility with the existing admission application.
Data layer	MySQL	Stores admission, registration, target, program, and historical datasets	Selected for relational data integrity, transaction support, indexing, and established integration with Laravel; PostgreSQL is a possible alternative.
Analytics layer	Python	Performs forecasting, risk scoring, warning generation, and recommendation processing	Selected for its extensive data analysis and machine learning libraries and its ability to operate as an independently scalable analytics service.
Forecasting model	Prophet	Predicts registration trends from time series admission data	Selected as an interpretable baseline for trend, seasonality, changepoint, and event modelling; Autoregressive Integrated Moving Average (ARIMA), Random Forest, and XGBoost will be evaluated as alternatives.
Integration	REST API	Transfers preprocessed data and predictive outputs between Laravel and Python	Selected for language independent communication, loose coupling, maintainability, and independent scaling of the application and analytics services.

Table 2 presents the proposed technology stack, system functions, and selection justifications for the Artificial Intelligence (AI) Business Intelligence architecture. The architecture consists of several layers where Vue.js in the presentation layer provides an interactive interface for displaying admission information and predictive results. Laravel in the application layer manages authentication, APIs, and business logic, while MySQL in the data layer stores structured admission and historical datasets. Python in the analytics layer performs forecasting, risk scoring, warning generation, and recommendation processing. The forecasting model applies Prophet due to its capability to handle trends, seasonality, and changepoints in time-series data. REST API integration connects Laravel and Python to enable modular communication, scalability, and maintainability. Overall, the proposed technology stack integrates data management, application services, and predictive analytics to support data-driven student admission decisions.

2.6. Planned Implementation and Validation Framework

Because this study focuses on architectural design, the proposed system has not yet been deployed using operational admission data, and no empirical performance claims are made. Future implementation will use de-identified longitudinal admission records, divided chronologically into training and testing periods to prevent information leakage. Rolling origin validation will be applied to assess model stability.

Forecasting performance will be evaluated using Mean Absolute Error, Root Mean Squared Error, and Mean Absolute Percentage Error. Prophet will be compared with ARIMA, Random Forest, and XGBoost using

identical data periods. REST API performance will be assessed through latency, throughput, failure rate, and concurrent request testing, while dashboard usability will be evaluated using task completion, completion time, error frequency, System Usability Scale, and user feedback. These evaluations will assess prediction reliability, technical performance, and usability in admission management.

3. RESULTS AND DISCUSSION

3.1. Admission Data Collection and Prediction Pipeline

The proposed architecture begins with an admission data collection and prediction pipeline. The pipeline describes how operational admission records are transformed into predictive outputs that can be displayed through dashboard components. The process starts from admission records, including lead data, applicant data, registration data, target data, program data, and historical trend data. These records are stored in the MySQL database and retrieved by the Laravel application layer for aggregation and preprocessing.

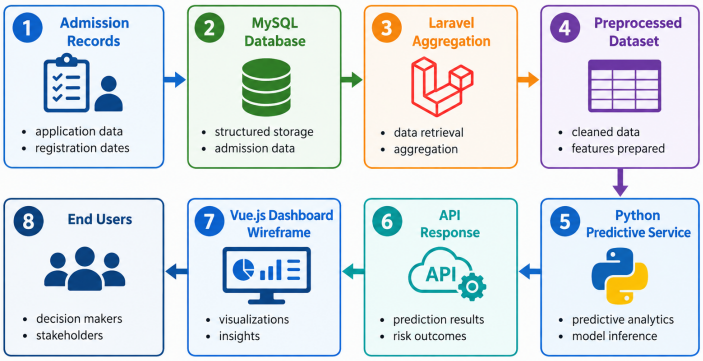


Figure 1. Admission Data Collection and Prediction Pipeline

Figure 1 illustrates the workflow of the proposed AI-driven Business Intelligence architecture, starting from admission record collection and storage in MySQL. The data are retrieved and aggregated through Laravel, preprocessed, and then analyzed using the Python predictive service. The prediction results are transferred through the Representational State Transfer (REST) API and displayed on the Vue.js dashboard as visualizations and insights. Finally, the generated insights are provided to decision-makers and stakeholders to support data-driven student admission planning.

The pipeline consists of several technical stages. Admission records are first collected from the existing admission system and stored in the MySQL database. Laravel then retrieves the data and aggregates them based on admission period, study program, registration status, and target achievement. The aggregated data are transformed into a preprocessed dataset that can be used by the Python predictive service. This dataset includes time series registration data for forecasting and program level data for risk classification.

After preprocessing, the dataset is sent to the Python predictive service. Prophet based forecasting is used to estimate registration trends, while risk scoring is used to classify study programs based on predicted achievement and target conditions. The predictive service also supports warning and recommendation generation through threshold based and rule based logic. The outputs are then returned to the Laravel API and rendered through Vue.js dashboard wireframe components. This pipeline shows that the proposed system is not limited to dashboard visualization, but includes a complete data processing and prediction workflow.

3.2. Existing Dashboard as Baseline System

The existing student admission dashboard is used as the baseline system for the proposed architecture. It provides the initial information environment for monitoring admission activities and serves as the foundation upon which the predictive functions are integrated. The dashboard already displays several descriptive indicators, such as prospective leads, applicants, registered students, admission targets, latest acquisition, previous period comparison, and program level distribution. These indicators provide an overview of the current admission situation and help users identify differences in performance across admission periods and study programs. These indicators support monitoring and reporting because they allow users to observe current admission per-

formance and compare it with institutional targets. The availability of these descriptive indicators also provides the necessary data context for developing additional predictive functions within the proposed architecture.

However, the existing dashboard does not yet include a predictive analytics service. Its primary function remains focused on presenting historical and current admission information rather than generating predictions or automated decision-support outputs. It displays stored admission data, but it does not perform forecasting, risk classification, warning generation, or recommendation logic. Consequently, the dashboard provides information that still requires further interpretation by users before appropriate managerial actions can be determined. As a result, users still need to interpret dashboard indicators manually before deciding whether a study program requires intervention. This manual interpretation may require users to compare multiple indicators and admission periods before identifying potential problems or emerging risks. For example, the dashboard may show that a program has low registration achievement, but it does not automatically estimate future registration trends or classify the program into a specific risk category. Therefore, the absence of predictive capabilities limits the dashboard's ability to provide early information about potential admission performance problems.

The proposed architecture treats the existing dashboard as a descriptive baseline that can be extended through predictive services. This approach preserves the existing dashboard functionality while introducing additional analytical capabilities to support more proactive admission management. Instead of replacing the current dashboard, the model adds a technical layer for data preprocessing, forecasting, risk scoring, alert processing, and recommendation generation. The additional layer processes existing admission data and transforms it into predictive outputs that can be integrated with the current dashboard interface. This approach allows the institution to improve dashboard capability gradually while maintaining the existing admission data structure. Furthermore, the modular approach enables predictive services to be developed and maintained independently, reducing the need for major changes to the existing information system. Overall, the proposed extension transforms the dashboard from a primarily descriptive monitoring tool into a predictive decision support environment that can assist users in identifying risks, anticipating future admission trends, and determining appropriate managerial actions.

3.3. Mapping of Data Inputs into Predictive Modules

The predictive workflow requires a clear mapping between data inputs and analytical modules to ensure that each admission dataset is processed according to its specific function. In the proposed architecture, time series registration data support forecasting, registration and target data enable achievement classification, program-level registration data facilitate risk scoring, previous period trends support early warning detection, and identified risk conditions are used for recommendation generation. Forecasting estimates future registration performance, while achievement classification compares predicted outcomes with admission targets. Risk scoring identifies programs requiring attention, early warning detects changes in registration trends, and the recommendation module generates appropriate managerial actions. Through the integration of these components, the workflow establishes a continuous process from data preparation and prediction to risk identification and decision support, enabling the transformation of admission data into actionable insights for systematic and data-driven student admission planning.

Table 3. Mapping of Data Inputs into Predictive Modules

Data Input	Predictive Module	Decision Rule or Threshold	Technical Output
Time series registration data	Prophet forecasting	Daily or weekly registration data are processed to estimate the final admission outcome	Predicted registration trend
Registration and target data	Target achievement module	$PAR_i = (F_i/T_i) \times 100\%$	Predicted achievement ratio
Program level registration data	Risk scoring module	High risk: $PAR_i < 60\%$; medium risk: $60\% \leq PAR_i < 80\%$; low risk: $PAR_i \geq 80\%$	Program risk category
Recent and previous period trends	Early warning module	The risk level increases when the recent trend declines and the forecast is below the previous period	Risk adjustment and trend alert
Risk and target conditions	Recommendation module	High risk: campaign intensification and applicant follow-up; medium risk: weekly monitoring; low risk: routine monitoring	Rule based managerial action

Table 3 presents the mapping of data inputs to predictive modules, decision rules, and their corresponding technical outputs. Time series registration data are processed through the Prophet forecasting module to generate predicted registration trends. Registration and target data are used by the target module to calculate the predicted achievement ratio, while program level registration data are analyzed through the risk scoring module to classify programs into high, medium, or low risk categories. Recent and previous-period trends are processed by the early warning module to identify potential declines in registration performance. Finally, risk and target conditions are used by the recommendation module to generate rule based managerial actions. Overall, the table shows how admission data are transformed into predictive outputs that support monitoring, risk identification, early warning, and managerial decision making.

This mapping shows that admission data are not only used for visualization, but also as computational inputs for predictive modules. The forecasting module estimates future registration trends based on historical and current registration data. The Target Achievement Module.interprets prediction results in relation to institutional or program targets. The risk scoring module classifies programs into risk categories, while the early warning module detects conditions that may require attention. The recommendation module then translates risk and target conditions into suggested actions.

The mapping also supports modular system development. Each predictive module can be developed and refined independently as long as the required input data are available. For example, the forecasting module can initially use Prophet and later be compared with Autoregressive Integrated Moving Average (ARIMA), Random Forest, or XGBoost. Similarly, the risk scoring and recommendation modules can be adjusted based on institutional requirements or admission policy changes.

3.4. Artificial Intelligence (AI) BI System Architecture

The proposed architecture separates presentation, application logic, data storage, and predictive analytics into modular layers. Admission data stored in MySQL are retrieved and prepared by Laravel before being transmitted through Representational State Transfer (REST) API to the Python analytics service. The resulting forecasts, risk categories, warnings, and recommendations are returned to the application layer and presented through Vue.js dashboard components. This separation allows the predictive model and interface components to be updated independently without redesigning the core admission system.

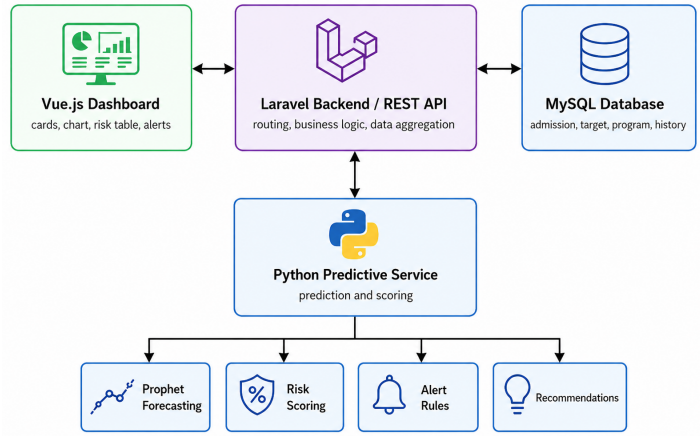


Figure 2. Web-Based AI-Driven BI System Architecture

Figure 2 illustrates the proposed web-based Artificial Intelligence (AI) Business Intelligence system architecture for student admission management. The Vue.js dashboard serves as the presentation interface for displaying admission cards, charts, risk tables, and alerts. The Laravel backend and Representational State Transfer (REST) API manage routing, business logic, and data aggregation while connecting the dashboard with the MySQL database, which stores admission, target, program, and historical data. The Python predictive service processes the data and performs forecasting and scoring using Prophet Forecasting, Risk Scoring, Alert Rules, and Recommendation modules. The integration of these components enables the system to transform admission data into predictive information and actionable insights to support data driven admission decision-making.

In this architecture, Vue.js is responsible for rendering dashboard wireframe components, including prediction summary cards, forecasting chart, program risk table, warning panel, and recommendation panel. The Vue.js interface sends requests to the Laravel backend to retrieve processed prediction outputs. Laravel manages routing, authentication, API endpoints, data aggregation, and communication between the database and predictive analytics service. MySQL stores structured admission data, including leads, applicants, registration records, targets, study programs, and historical trends.

The Python analytics layer receives preprocessed admission data from Laravel through Representational State Transfer (REST) API. In this layer, Prophet is used to perform time series forecasting based on historical and current registration data. The forecasting results are then used to support risk scoring, alert generation, and recommendation logic. After the predictive processing is completed, the Python service returns structured outputs to Laravel. These outputs are then rendered by Vue.js into dashboard components.

This architecture supports modular implementation because the predictive analytics layer is separated from the main application layer. Laravel can continue to manage the core admission application, while Python can be developed as an independent analytics service. This structure also allows future model improvement because Prophet can be compared with or replaced by other algorithms such as Autoregressive Integrated Moving Average (ARIMA), Random Forest, or XGBoost without redesigning the entire dashboard system.

3.5. Predictive Dashboard Wireframe Structure

The dashboard wireframe translates predictive outputs into five functional areas: summary indicators, forecasting visualization, program risk classification, warning alerts, and recommended actions. The component based structure supports independent development and testing while maintaining a consistent flow from prediction results to managerial interpretation.

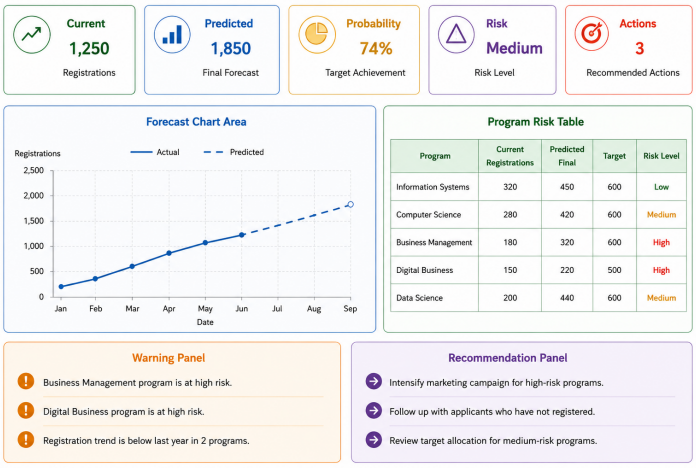


Figure 3. Predictive Dashboard Wireframe Structure

Figure 1-3 illustrates the proposed predictive dashboard wireframe for supporting student admission decision making. The dashboard provides key indicators including current registrations, predicted final registrations, target achievement probability, risk level, and recommended actions. The Forecast Chart Area presents actual and predicted registration trends, while the Program Risk Table compares registration targets, predictions, achievement ratios, and risk levels across study programs. The Warning Panel highlights potential admission problems that require attention, whereas the Recommendation Panel provides suggested managerial actions for addressing identified risks. Overall, the wireframe integrates forecasting, risk assessment, warnings, and recommendations into a single interface to support faster and more data driven admission decisions.

The wireframe consists of several component zones. The prediction summary cards display key outputs such as current registration, predicted registration, target probability, risk status, and recommended actions. The forecast chart area displays actual registration, predicted registration, and target lines so users can compare current progress with expected outcomes. The program risk table presents program level outputs such as predicted achievement, risk category, and intervention priority. The warning panel displays alerts generated from threshold based rules, while the recommendation panel displays rule based action suggestions.

From a frontend perspective, the wireframe can be implemented using reusable Vue.js components. The prediction summary cards can be developed as PredictionCard, the chart area as ForecastChart, the risk table as RiskTable, the warning section as WarningPanel, and the recommendation section as RecommendationPanel. Each component receives data from Laravel API endpoints and updates when new prediction outputs are available. This component based design improves maintainability because each dashboard section can be developed, tested, and modified independently.

3.6. Architecture Centered Discussion

The principal architectural contribution is the separation of operational admission management from predictive processing. This modular design allows forecasting and risk analysis capabilities to be integrated incrementally without replacing the existing information system. It also enables predictive models to be evaluated, updated, or replaced independently from the application and presentation layers, improving maintainability and long term flexibility.

The proposed architecture integrates data storage, preprocessing, predictive analytics, Representational State Transfer (REST) API communication, and dashboard rendering within a unified workflow. Unlike a conventional dashboard that only displays historical data, the system also presents outputs generated by forecasting and risk classification modules. This integration enables decision makers to access both operational and predictive insights through a single interface.

Laravel manages application logic, user authentication, routing, database interaction, and Representational State Transfer (REST) API endpoints, while Python performs forecasting and analytical computation. This separation of responsibilities improves maintainability and allows analytical models to evolve independently from the operational system. As a result, new predictive capabilities can be introduced with minimal impact on existing application services.

Prophet is positioned as an interpretable baseline forecasting model and should be compared with Autoregressive Integrated Moving Average (ARIMA), Random Forest, and XGBoost using identical chronological data partitions, forecasting horizons, and evaluation metrics before selecting the production model. The Representational State Transfer (REST) API architecture also supports scalability by allowing the analytics layer to be optimized or expanded independently as data volume and predictive requirements increase.

Overall, the proposed architecture provides a technical pathway for transforming descriptive admission dashboards into predictive decision support systems through forecasting, risk scoring, warning generation, and recommendation logic. However, these findings represent architectural and design contributions rather than empirical evidence of forecasting accuracy, API performance, or dashboard usability. Therefore, the proposed data flow, system integration, and interface shown in Figures 3 should be validated using real admission records and evaluations involving actual system users.

4. MANAGERIAL IMPLICATIONS

The proposed Artificial Intelligence (AI) BI architecture provides practical support for admission managers by enabling earlier monitoring of target achievement. Through forecasting outputs, risk categories, warning alerts, and recommendation logic, managers can identify whether current registration trends are sufficient to reach admission targets before the admission period ends. This reduces dependence on manual interpretation of dashboard indicators and supports more timely intervention. For marketing teams and study programs, the architecture helps prioritize intervention based on predicted risk. Programs with high risk status can be targeted for stronger digital campaigns, applicant follow-up, communication improvement, or target review. This allows recruitment resources to be allocated more selectively instead of applying the same strategy to all study programs. For system developers, the architecture provides a gradual implementation pathway in which predictive functions can be introduced as an independent service. This reduces migration risk, supports maintainability, and allows future improvements to the forecasting and decision rule modules.

5. CONCLUSION

This study proposes an Artificial Intelligence (AI) Business Intelligence architecture for predictive student admission decision support in higher education. Its main contribution is an integrated and modular workflow that transforms admission data into forecasts, program level risk classifications, warning alerts, and recommended actions. The architecture provides a practical pathway for extending descriptive admission

dashboards into predictive decision support systems without replacing the institution's existing information infrastructure.

The proposed model integrates data preprocessing, Prophet based forecasting, risk scoring rules, Representational State Transfer (REST) API communication, and reusable dashboard components. This structure enables admission managers to identify programs at risk of missing recruitment targets and undertake more focused interventions based on predicted outcomes rather than relying solely on historical and current indicators.

The principal limitation is that the architecture remains at the conceptual design stage and has not been evaluated using operational admission data. Therefore, the study does not report forecasting accuracy, API performance, system scalability, or dashboard usability. Future research should implement the architecture using de identified longitudinal datasets, compare Prophet with Autoregressive Integrated Moving Average (ARIMA), Random Forest, and XGBoost using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan Mean Absolute Percentage Error (MAPE). evaluate API latency, throughput, concurrency, and failure rates, and assess dashboard usability through task based testing and the System Usability Scale. These evaluations are necessary before the architecture can be recommended for full operational deployment.

6. DECLARATIONS

6.1. About Authors

Mohamad Rakhmansyah (MR) https://orcid.org/0009-0007-9392-3777

Untung Rahardja (UR) https://orcid.org/0000-0002-2166-2412

Muhammad Faizal Fazri (MF) https://orcid.org/0009-0002-9769-0257

Nikita Ivanov (NI) -

6.2. Author Contributions

Conceptualization: MR and UR Methodology: MR and MF; Software: MR and MF; Validation: UR and MF; Formal Analysis: MR and MF; Investigation: MR; Resources: UR; Data Curation: MR and MF; Original Draft Preparation: MR and MF; Review and Editing: UR, MF, and NI; Manuscript Processing: NI; All authors, MR, UR, MF, and NI, have read and agreed to the published version of the manuscript.

6.3. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.4. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.5. Declaration of Conflicting Interest

The authors declare that they have no conflicts of interest, known competing financial interests, or personal relationships that could have influenced the work reported in this paper.

REFERENCES

[1] N. P. Rana, S. Chatterjee, Y. K. Dwivedi, and S. Akter, "Understanding dark side of artificial intelligence (ai) integrated business analytics: assessing firm's operational inefficiency and competitiveness," *European Journal of Information Systems*, vol. 31, no. 3, pp. 364–387, 2022.

[2] A. Kanivia, H. Hilda, A. Adiwijaya, M. F. Fazri, S. Maulana, and M. Hardini, "The impact of information technology support on the use of e-learning systems at university," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 4, no. 2, pp. 122–132, 2024.

[3] K. Samuel, "Artificial intelligence in big data visualization: Advancing dashboard technology," *International Journal of Computer Science and Data Engineering*, vol. 1, no. 2, pp. 1–6, 2024.

[4] N. Lutfiani, N. P. L. Santoso, R. Ahsanitaqwim, U. Rahardja, and A. R. A. Zahra, "Ai-based strategies to improve resource efficiency in urban infrastructure," *International Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 121–127, 2024.

[5] S. Caruso, M. Bruccoleri, A. Pietrosi, and A. Scaccianoce, "Artificial intelligence to counteract "kpi overload" in business process monitoring: the case of anti-corruption in public organizations," *Business process management journal*, vol. 29, no. 4, pp. 1227–1248, 2023.

[6] U. Rahardja, "Using highchart to implement business intelligence on attendance assessment system based on yii framework," *International Transactions on Education Technology (ITEE)*, vol. 1, no. 1, pp. 19–28, 2022.

[7] I. K. Himmy'az, S. Pranata, M. Yusup, H. Kusumah, J. Zanubiya, E. A. Natalia *et al.*, "A blockchain based framework for improving data quality and predictive accuracy in business intelligence systems," in *2025 4th International Conference on Creative Communication and Innovative Technology (ICCIT)*. IEEE, 2025, pp. 1–7.

[8] R. Sultana, "Artificial intelligence in data visualization: Reviewing dashboard design and interactive analytics for enterprise decision-making," *International Journal of Business and Economics Insights*, vol. 5, no. 3, pp. 01–29, 2025.

[9] N. S. Aljerjawi and S. S. Abu-Naser, "Reinventing classical sorting with deep learning and reinforcement techniques," *International Journal of Academic Information Systems Research (IJAISR)*, vol. 9, no. 6, pp. 126–133, 2025.

[10] D. Novitasari and F. C. Young, "Harnessing technology and data analytics for marketing reinvention," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 6, no. 1, pp. 92–103, 2026.

[11] E. A. Natasya, N. P. L. Santoso, L. Pasha, C. T. Hua, and C. Perez, "Enhancing adaptive learning environments in learning factories through artificial intelligence," *International Transactions on Education Technology (ITEE)*, vol. 4, no. 1, pp. 1–13, 2025.

[12] C. V. Ibeh, O. F. Asuzu, T. Olorunsogo, O. A. Elufioye, N. L. Nduubuisi, A. I. Daraojimba *et al.*, "Business analytics and decision science: A review of techniques in strategic business decision making," *World Journal of Advanced Research and Reviews*, vol. 21, no. 2, pp. 1761–1769, 2024.

[13] E. J. A. H. Nasution, "Digital governance model for zakat based on mui fatwas in indonesia," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 5, no. 2, pp. 223–234, 2025.

[14] M. M. I. Jabed, A. S. Khawer, S. Ferdous, D. H. Niton, A. B. Gupta, and M. S. Hossain, "Integrating business intelligence with ai-driven machine learning for next-generation intrusion detection systems," *International Journal of Research and Applied Innovations*, vol. 6, no. 6, pp. 9834–9849, 2023.

[15] M. N. Ayubi, S. Anggoro, and Y. M. Kareem, "Decentralized decision intelligence using ai and blockchain in modern enterprises," *ADI Journal on Recent Innovation*, vol. 7, no. 1, pp. 38–49, 2025.

[16] N. Lutfiani, S. Wijono, U. Rahardja, A. Iriani, Q. Aini, and R. A. D. Septian, "A bibliometric study: Recommendation based on artificial intelligence for ilearning education," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, no. 2, pp. 109–117, 2023.

[17] N. I. Susanthi, M. Ali, and A. H. Hernawan, "Digital learning platforms as facilitator for university-business collaboration in logistics management curriculum design," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 6, no. 1, pp. 37–50, 2026.

[18] H. Handoko, S. B. Waluya, W. Wardono, S. Sugiman, and L. W. Ming, "Geometry learning pathways for creative thinking via edupreneurial approaches," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 8, no. 2, pp. 669–681, 2026.

[19] R. Septiowati, "Analysis of the application information technology on employee work productivity pt rajendra kesatria perkasa depok," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 6, no. 1, pp. 1–9, 2026.

[20] H. D. Natalia and A. Aprillia, "Exploring the impact of e-wom on generation z purchase intention: The mediating role of brand image and perceived quality," *IAIC Transactions on Sustainable Digital Innovation (ITSDI*, vol. 6, no. 2, pp. 164–176, 2026.

[21] S. Saibaba, "Profiling critical thinking and ai attitudes in management education using a factor-cluster approach for the digital business era," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 8, no. 2, pp. 493–507, 2026.

[22] U. Arumugam and M. Manida, "Enhancing strategic decision making: Assessing the role and impact of artificial intelligence in business management," *Journal of Business Analytics and Data Visualization*, vol. 5, no. 1, pp. 1–8, 2024.

[23] R. Indrawan, E. D. Very, D. Tribuana, and E. A. Nabila, "Aiot driven smart solar system for real time predictive sustainable energy management," *International Transactions on Artificial Intelligence*, vol. 4,

no. 1, pp. 105–114, 2025.

[24] S. Vudugula, S. K. Chebrolu, M. Bhuiyan, and F. Z. Rozony, "Integrating artificial intelligence in strategic business decision-making: A systematic review of predictive models," *International Journal of Scientific Interdisciplinary Research*, vol. 4, no. 1, pp. 01–26, 2023.

[25] W. W. A. Effendi, M. Maswanto, M. Murod, and I. Maria, "Artificial intelligence enhancing financial regulatory compliance through regtech governance applications," *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, vol. 7, no. 2, pp. 216–225, 2026.

[26] L. Limajatini, S. Suhendra, G. A. Pangilinan, and M. G. Ilham, "Integration of artificial intelligence in the financial sector innovation, risks and opportunities," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 5, no. 1, pp. 58–70, 2025.

[27] K. I. M. Dewi, I. W. G. Narayana, and R. L. Rahardian, "Application of certification management information systems at lsp engineering hospitality indonesia," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, no. 3, pp. 227–239, 2023.

[28] D. Deng, A. Wu, H. Qu, and Y. Wu, "Dashbot: Insight-driven dashboard generation based on deep reinforcement learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 690–700, 2022.

[29] A. Manawar, "An innovative and secure platform for leveraging the blockchain approach for online exams," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, no. 1, pp. 99–108, 2023.

[30] Q. Aini, D. Manongga, U. Rahardja, I. Sembiring, and Y.-M. Li, "Understanding behavioral intention to use of air quality monitoring solutions with emphasis on technology readiness," *International Journal of Human–Computer Interaction*, vol. 41, no. 8, pp. 5079–5099, 2025.

[31] N. Rane, M. Paramesha, S. Choudhary, and J. Rane, "Business intelligence and business analytics with artificial intelligence and machine learning: Trends, techniques, and opportunities," *Techniques, and Opportunities (May 17, 2024)*, 2024.