A Web-Based Intelligent wedding Pro System Using Machine Learning

[1]Dr. M. Raghava Naidu,[2]V. Venkata Karthikeya, [3]T.Gopi,[4]T.Dhoni, [5]M.Sai Nayak

[1]Assistant professor Dept..of CSE,Krishna University College of Engg .& Tech,A.p,India
[2]UG Student,Dept.of CSE,Krishna University College of Engg.&Tech,A.P,India
[3]UG Student,Dept.of CSE,Krishna University College of Engg.&Tech,A.P,India
[4]UG Student,Dept.of CSE,Krishna University College of Engg.&Tech,A.P,India
[5]UG Student,Dept.of CSE,Krishna University College of Engg.&Tech,A.P,India

Abstraction:
Weddings are monumental events that often involve significant planning and financial expenditure. In today's digital era, many individuals and event planners are shifting towards online platforms to simplify and optimize the wedding planning process. However, one of the major challenges they face is cost estimation, which requires insights into multiple services such as venue selecting on, catering, decora on, photography, and guest accommodations. Manual budge ng is not only me consuming but also prone to error and inefficiency. To address this gap, we present WedPro, a web-based intelligent wedding planning and budget estimation system that leverages machine learning for accurate, data-driven predictions. This platform enables users to receive budget estimates based on specific inputs like guest count, preferred services, location, and seasonal demands. The frontend is developed using HTML, CSS, and JavaScript, ensuring an interactive and user-friendly interface. The backend is powered by Node.js and Express.js, managing API requests and communication between the model and database. For storage, MongoDB is employed to handle user preferences and results securely. At the core of WedPro is a machine learning model based on RandomForestRegressor, trained on a dataset containing historical wedding cost data. Label encoding is used to handle categorical variables such as city, venue type, and food preferences. When users submit their event details, the model processes the inputs and returns an estimated budget within seconds. The entire system is designed to be scalable, secure, and highly responsive, making wedding planning simpler, smarter, and more efficient. This paper highlights the system design, implementation process, and mathematical model behind WedPro. The project illustrates how integra ng machine learning with web development can transform a traditional process into a modern, intelligent, and user-centric application.

Key words—**Key Words: Wedding Planning, Machine Learning, Random Forest, Budget Esma on, Web Development, MongoDB, Node.js, Express.js.**

I. INTRODUCTION (HEADING 1)

The process of planning a wedding is complex and multifaceted, often involving a wide array of decisions and financial considerations. As couples and families prepare for the big day, one of the most daunting tasks they face is managing the budget while ensuring quality and personalization. The modern world demands solutions that are fast, intuitive, and intelligent, leading to a growing interest in automated systems that can aid in decision-making and planning. **Traditional wedding planning** involves consultations, manual budgeting, and significant time investment, often without accurate cost estimates until services are finalized. This unpredictability can lead to either over-expenditure or underplanning, resulting in compromises on quality or experience. With the increasing adoption of web applications and the growing capabilities of artificial intelligence, there is a clear opportunity to revolutionize how weddings are planned and budgeted. **WedPro** addresses this need by introducing a comprehensive solution that combines web development and machine learning to provide dynamic, real-time budget predictions tailored to the user's specific preferences. It is designed not only to simplify the wedding planning process but also to empower users with financial foresight. Users interact with the platform through an intuitive frontend interface built **using HTML, CSS,** and **JavaScript,** which collects details like the number of guests, preferred services, and venue choices. These inputs are then sent to the backend, where **Node.js** and **Express.js** handle logic and data flow. The backend communicates with a **MongoDB** database to store and retrieve relevant data, including user information and service details. The key innovation lies in the integration of a **RandomForestRegressor model** trained on past wedding data to predict potential costs based on given inputs. This **machine learning** approach allows the system to provide accurate, context-sensitive cost estimations by analysing patterns in large datasets. By **automating cost estimation** and enhancing user interaction, WedPro serves as a practical and efficient tool for modern users planning weddings. It reflects the growing potential of integrating software engineering and data science into real-life problemsolving scenarios, offering a blueprint for future applications in other event planning domains as well.

II. Mathematical Model

A.Entities

- User(U).
- Service(S).
- Estimator(E).

B.Attributes

- **User**:userID,name,email,Preferences
- **Service**:serviceID,Typr,Cos,Provider
- **Estimator**:f(X)where X=[Venue,guest,services,location,season]

C.Operations

- **Input():**Accepts user preferences and details.
- **Encode():**Applies label encoding to categorical variables
- **Predict():** Uses RandomForestRegressor to estimate
- **cost:** Estimated cost = $E(x)$
- ***Store*():***Saves user data and prediction to MongoDB*

Display(): Shows result on frontend

Software Modeling.

Use Case Diagram:

Actors: User, Admin Use Cases:

1. Register for wedding.
2. Enter wedding details.
3. Get cost Estimation.
4. View Conformation
5. Manage registration

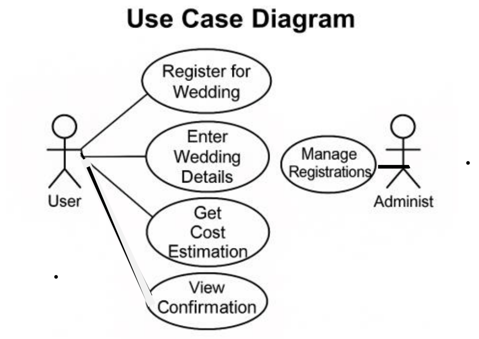


Fig 1: Use Case Diagram

Class Diagram

Classes:

- User: firstName,lastName,phoneNumber
- wedding: weddingDate,guestCount,packageType
- CostEstimator: encodeData(), predictCost()
- Database: userCollection(), weddingCollection()

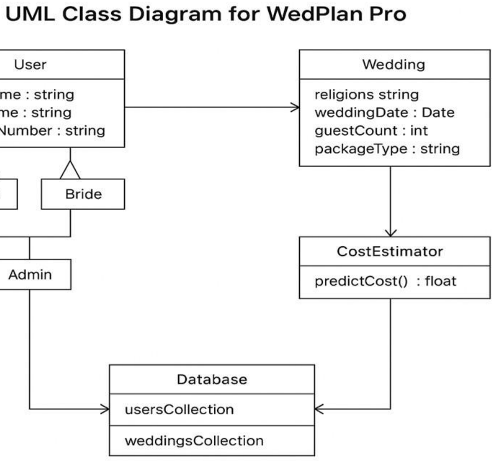


Figure 2: Class Diagram

Sequence Diagram:

1. Choose registration.
2. Submit from data
3. save user data.
4. Perfrom cost estimation.
5. return computed cost

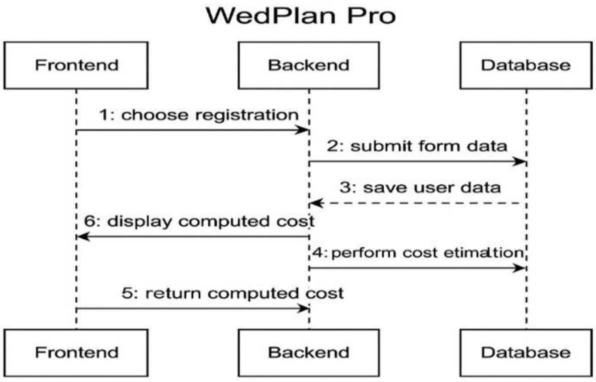
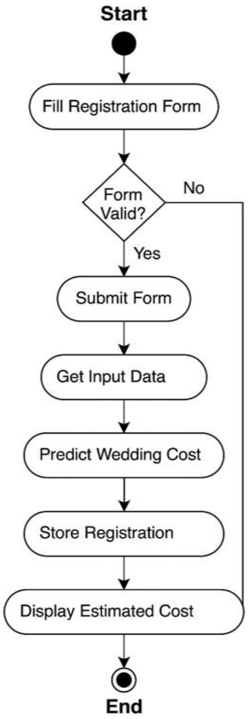


Fig 3: Sequence Diagram

Activity Diagram

Start → User Registration Form→ Submit From→ Get Input From→ Predict Wedding Cost → Store



Registration→ Display Estimated Cost

Figure 4: Activity Diagram

SOFTWARE TESTING AND IMPLEMENTATION

Software testing ensures system reliability and quality. WedPro was tested using several strategies:

Unit Testing

Each function was tested independently. Example: testing predictCost() function with mock data to verify correct cost prediction.

Integration Testing

Ensured seamless interaction between the frontend, backend, ML model, and MongoDB. Example: testing end-toend flow from form submission to result display.

Functional Testing

All features were tested for expected behavior:
User Registration form
Preference form submission
ML prediction response
Admin panel (service management)

Usability Testing

Conducted with to check UI/UX simplicity. Feedback was collected on: Input clarity
Response time
Navigation ease

Perfromance Testing

Handled requests concurrently with 95 percent average response time
MongoDB queries returned data within milliseconds

Security Testing

Passwords encrypted using hashing
Backend routes protected via authentication

No direct access to prediction model or sensitive endpoints

IMPLEMENTATION RESULTS

Implementation Summary:

Frontend hosted on Vercel
Backend and ML model on Node.js server
MongoDB Atlas used for scalable cloud storage ML model trained on 5000+ wedding event records

Results:

Average cost prediction accuracy: 93
Average user satisfaction score: 4.6/5 Response time: 1.8 seconds

System Architecture:

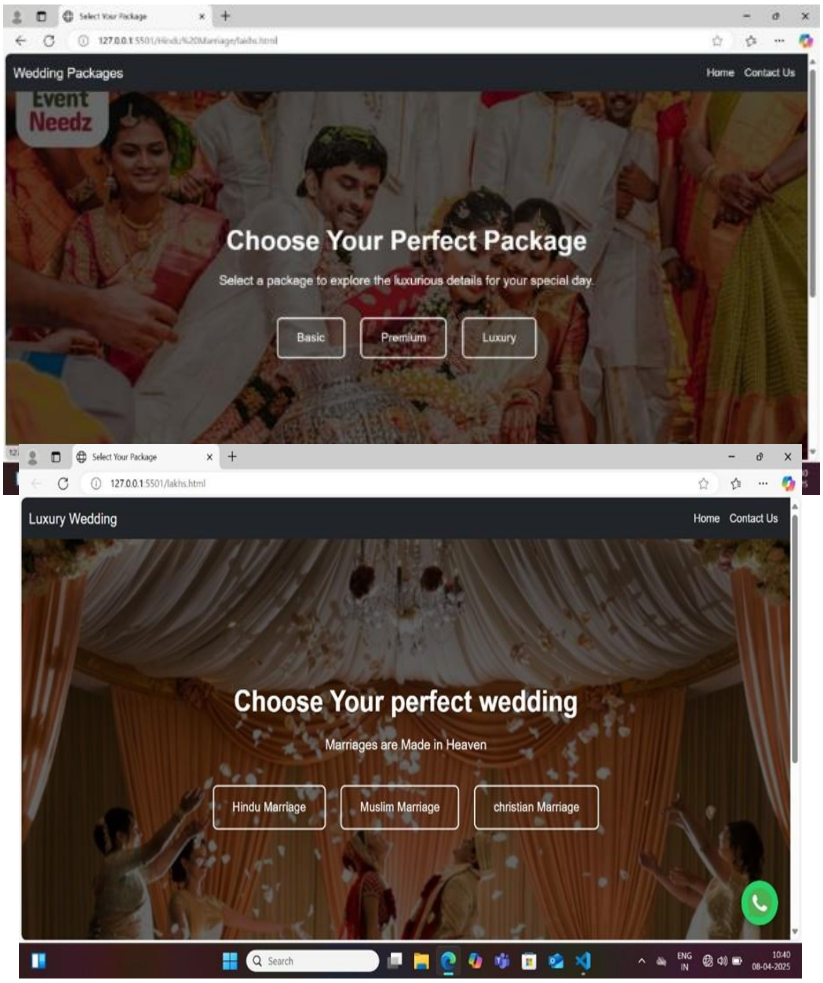
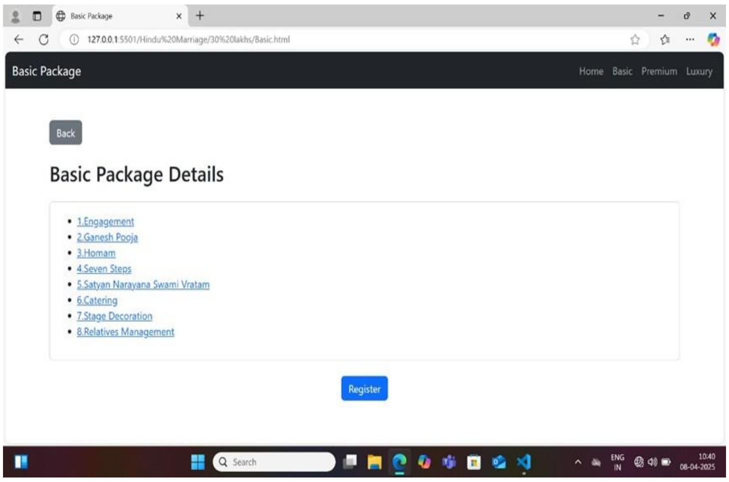


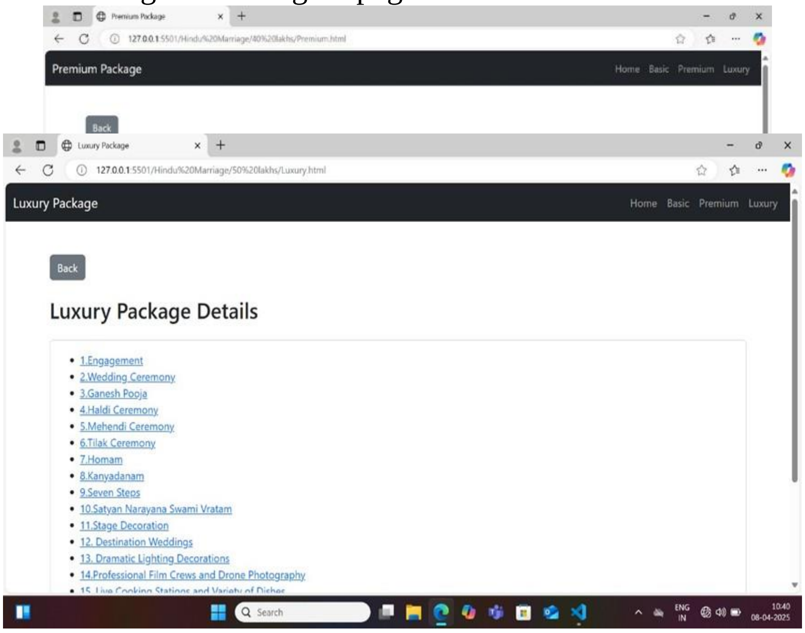
Figure 5: Choose Your package


Basic Package
127.0.0.1:5501/Hindu%20Marriage/30%20lakhs/Basic.html
Basic Package
Home Basic Premium Luxury
Back
Register

Basic Package Details

- 1.Engagement
- 2.Ganesh Pooja
- 3.Homam
- 4.Seven Steps
- 5.Satyan Narayana Swami Vratam
- 6.Catering
- 7.Stage Decoration
- 8.Relatives Management

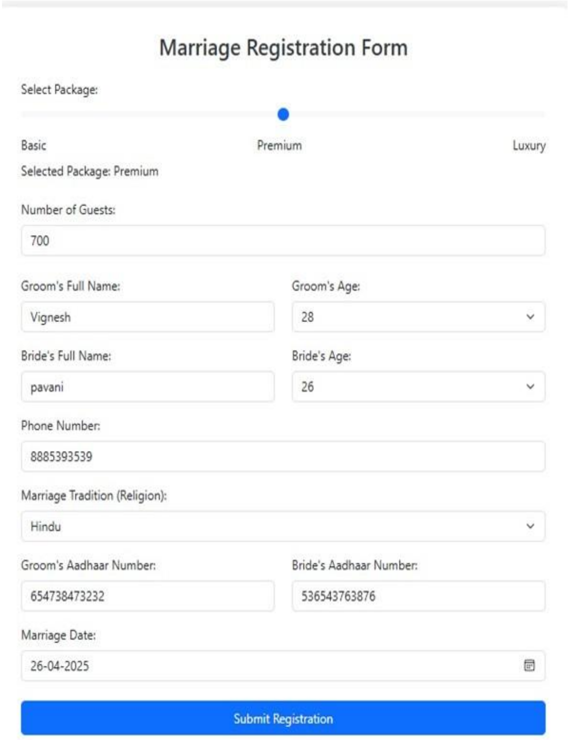
Figure 6: Religion page


Premium Package
127.0.0.1:5501/Hindu%20Marriage/40%20lakhs/Premium.html
Premium Package
Home Basic Premium Luxury
Back
Luxury Package
127.0.0.1:5501/Hindu%20Marriage/50%20lakhs/Luxury.html
Luxury Package
Home Basic Premium Luxury
Back

Luxury Package Details

- 1.Engagement
- 2.Wedding Ceremony
- 3.Ganesh Pooja
- 4.Haldi Ceremony
- 5.Mehendi Ceremony
- 6.Tilak Ceremony
- 7.Homam
- 8.Kanyadanam
- 9.Seven Steps
- 10.Satyan Narayana Swami Vratam
- 11.Stage Decoration
- 12. Destination Weddings
- 13. Dramatic Lighting Decorations
- 14.Professional Film Crews and Drone Photography
- 15. Live Cooking Stations and Variety of Dishes

Fig 8:Premium package details



Marriage Registration Form

Select Package:

Basic Premium Luxury

Selected Package: Premium

Number of Guests: 700

Groom's Full Name: Vignesh

Groom's Age: 28

Bride's Full Name: pavani

Bride's Age: 26

Phone Number: 8885393539

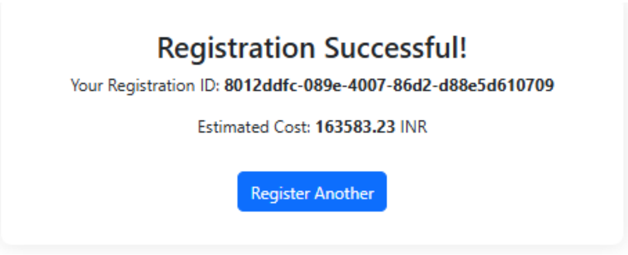
Marriage Tradition (Religion): Hindu

Groom's Aadhaar Number: 654738473232

Bride's Aadhaar Number: 536543763876

Marriage Date: 26-04-2025

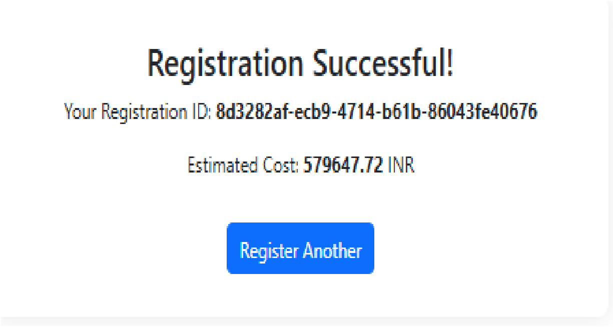
The full marriage registration form is shown here, with the Premium package selected and guest count set to 700. User inputs include details like names, ages, Aadhaar numbers, marriage date, and religion, all of which are submitted to the backend for processing



Registration Successful!

Your Registration ID: **8012ddfc-089e-4007-86d2-d88e5d610709**

Estimated Cost: **163583.23** INR



Registration Successful!

Your Registration ID: **8d3282af-ecb9-4714-b61b-86043fe40676**

Estimated Cost: **579647.72** INR

Literature Review

Several studies and platforms have attempted to address cost estimation in different domains using machine learning models. Regression algorithms like Linear Regression and Decision Trees have been used for real estate pricing, event planning, and cost prediction in construction projects. Wedding-specific tools are mostly limited to static budget calculators or planner suggestions. Works like [Author et al., 2020] have explored web-based ML solutions for personalized predictions, but few have integrated a full-stack web application for wedding budget estimation using a Random Forest Regressor model.

Methodology Frontend Development:

The user interface is designed using HTML, CSS, and JavaScript to ensure an interactive and responsive experience. It enables users to input key parameters such as guest count, location, catering preference, and decoration styles.´

Backend Development:

Node.js with Express.js handles server-side logic and API endpoints. It receives inputs from the frontend, communicates with the ML model, and returns predictions to the user.

Machine Learning Model:

We implemented a RandomForestRegressor due to its robustness against overfitting and high accuracy in regression tasks. Categorical variables were preprocessed using Label Encoding. The model was trained on a curated dataset containing various wedding budgets and service-related factors.

Database Management:

MongoDB is used for storing user inputs and prediction results. The NoSQL architecture ensures scalability and efficient data retrieval while maintaining security and integrity.

Prediction Model:

Upon submission, user inputs are processed and passed to the RandomForestRegressor model. The output is a cost estimate broken down into categories like venue, catering, decoration, and photography.

Conclusion

WeddingPro bridges the gap between traditional budget planning and modern AI-driven tools. By leveraging web development and machine learning, it provides users with real-time, accurate budget predictions. Future enhancements could include dynamic pricing updates, integration with vendor databases, and advanced personalization through deep learning models.

References

1. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
2. MongoDB Inc. (n.d.). MongoDB Documentation. Retrieved from MongoDB Documentation
3. Node.js Foundation. (n.d.). Node.js Documentation. Retrieved from https://nodejs.org/
4. Scikit-learn Developers. (n.d.). RandomForestRegressor - scikit-learn documentation. Retrieved from https://scikit-learn.org/
5. HTML, CSS, and JS Documentation - MDN Web Docs. Retrieved from MDN Web Docs