© Volume 2, Issue 10, October 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

Vedic Astrology Fortune App: Design and Implementation of a Web-Based Kundli, Horoscope and Panchang Platform

[1]Janhavi Singh, [2]Supriya Singh, [3]Jay Chand
[1,2,3]*Kamla Nehru Institute of Physical & Social Science Engineering Institute, Faridipur, Sultanpur, Uttar Pradesh 228119, India*

***Abstract*—Millions of people in India and abroad use Vedic astrology (Jyotish) to understand birth charts, daily planetary effects, auspicious timings and marriage compatibility. Most existing apps are either too complex for beginners or are heavy with advertisements and unclear calculations. This paper presents the design and implementation of vedic-astrology-fortune.app, a simple and privacy-aware platform that generates a Kundli (birth chart), daily horoscope, Panchang and Guna Milan (compatibility score) from a person's date, time and place of birth. The system uses a layered architecture with a REST interface, four independent service modules and an ephemeris-based calculation engine that works in the sidereal zodiac with the Lahiri ayanamsa. We describe the calculation workflow, database design, user interface decisions, testing plan and ethical limits of the application. The paper also states clearly that astrological output is cultural and entertainment content, not scientific prediction. The proposed design is modular, easy to extend and suitable for students and small teams who wish to build a reliable astrology application.**

***Index Terms*—Vedic astrology, Kundli, Panchang, horoscope, Guna Milan, ephemeris, web application, REST API**

I. INTRODUCTION

Astrology has been part of Indian life for thousands of years. In the Vedic tradition, called Jyotish, a person's life is studied through the positions of the nine planets (Navagraha) and the twelve zodiac signs at the exact time and place of birth. Families still consult a Kundli before naming a child, fixing a marriage or choosing a date for an important event.

Earlier, people had to visit an astrologer for even a basic chart. Today, smartphones and web browsers allow the same chart to be created in seconds. However, many astrology apps have three common problems. First, the screens are crowded and use difficult Sanskrit terms without

explanation. Second, the method of calculation is hidden, so users cannot tell whether the chart is correct. Third, birth details are often collected without clear privacy rules.

This project, vedic-astrology-fortune.app, tries to solve these problems. The aim is to give an easy and honest tool that (i) calculates a correct sidereal chart, (ii) explains every result in plain language, and (iii) respects the privacy of the user. The main contributions of this paper are:

1) a modular, layered design for a Vedic astrology application;
2) a clear calculation workflow from birth data to Kundli, Panchang and Guna Milan;
3) a beginner-friendly interface plan with simple explanations; and
4) a discussion of testing, privacy and ethical limits.

The rest of the paper is organised as follows. Section II gives background. Section III states the problem and objectives. Sections IV and V explain the design and implementation. Section VI describes testing. Section VII covers privacy and ethics, Section VIII gives future scope and Section IX concludes.

II. BACKGROUND AND RELATED WORK

A. Basics of Vedic Astrology

Vedic astrology uses the sidereal zodiac, which is fixed against the background stars. Western astrology mostly uses the tropical zodiac, which is fixed to the seasons. Because of the slow wobble of the Earth's axis, the two differ by an angle called the ayanamsa, which is about 24 degrees at present [7]. Most Indian astrologers use the Lahiri (Chitrapaksha) ayanamsa.

Some key terms used in this paper are as follows. The Rashi is one of the 12 signs of 30 degrees each. The Lagna (ascendant) is the sign rising on the eastern horizon at birth. The Nakshatra is one of 27 lunar mansions of 13°20′ each. The Dasha is a planetary time period; the Vimshottari system divides a 120-year life cycle among nine planets [2]. The Panchang is the Hindu almanac with five parts: Tithi, Vara, Nakshatra, Yoga and Karana.

B. Existing Applications

Popular astrology apps offer horoscopes, chat with astrologers and paid reports. Open tools such as the Swiss Ephemeris [3] provide very accurate planetary positions and are used by many professional programs. The Meeus algorithms [1] are another well-known source for computing the Sun, Moon and planets. Our work does not invent new astronomy. Instead, it focuses on packaging these trusted calculation sources into a clean and understandable application.

C. Comparison with Typical Apps

Table I compares common features of typical commercial astrology apps with the approach proposed in this paper. The comparison is based on general observation of widely used apps and is meant to show design intent, not to criticise any product.

TABLE I TYPICAL APPS VERSUS PROPOSED APP

Feature	Typical apps	Proposed app
Explanations	Technical terms, little help	Plain language and tooltips
Calculation	Method not shown	Ayanamsa and method stated
Data use	Often unclear	Minimum data, easy deletion
Interface	Crowded, many pop-ups	Four-field form, clean pages
Claims	Strong predictions	Guidance and entertainment only

D. Cultural and Social Relevance

For many families, a Kundli is more than a prediction; it is a part of tradition, much like a family tree or a festival calendar. Young users often want to learn what these customs mean but find printed books hard to follow. A well-designed digital tool can therefore act as a gentle learning aid. It can explain why the Moon's position matters for the Rashi, how the 27 Nakshatras are formed, and why the Panchang is used to choose dates. In this way the application supports cultural awareness while still leaving the final belief and choice with the user.

III. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed is: how can a small development team build a Vedic astrology application that is accurate, easy to use for ordinary people and respectful of personal data?

A. Objectives

The objectives of the project are as follows.

1) Generate a Kundli (North and South Indian styles) from date, time and place of birth.
2) Provide a daily, weekly and monthly horoscope based on the Moon sign.
3) Show the daily Panchang for the user's location.
4) Calculate Guna Milan out of 36 points for two people.
5) Keep the interface simple, fast and available in English and Hindi.
6) Store the minimum personal data and let the user delete it at any time.

B. Requirements

The functional requirements follow directly from the objectives above. In addition, the following non-functional requirements guide the design.

Accuracy: calculated positions must agree with a trusted ephemeris.

Performance: a chart should appear within a few seconds on a normal mobile connection.

Usability: a first-time user should complete the form without any help.

Security: personal data must be protected during transfer and storage.

Maintainability: modules should be independent so that features can be added easily.

IV. SYSTEM DESIGN

A. Architecture

The application follows a layered architecture, as shown in Fig. 1. The user interface talks only to the REST API layer [5]. The API validates input and forwards it to one of four services. All services use one shared ephemeris engine for planet positions and one database for stored data. This separation means that a change in one part, for example a new chart style, does not break the others [4].

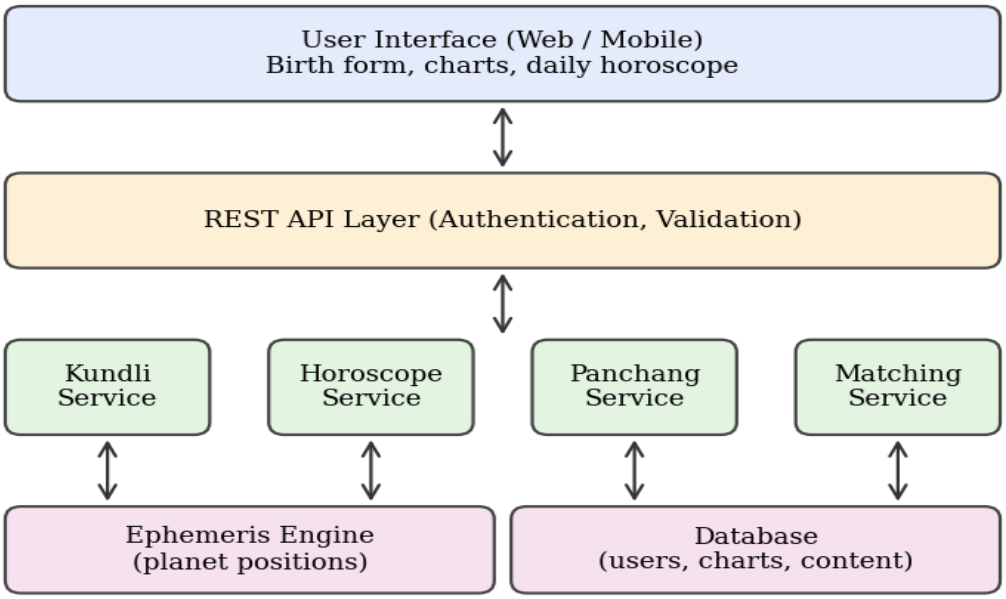

Fig. 1. Layered architecture of the proposed application.

B. Functional Modules

Table II lists the four functional modules with their inputs and outputs.

TABLE II MAIN MODULES OF THE APPLICATION

Module	Input	Output
Kundli	Date, time, birth place	Lagna, planets in houses, Nakshatra, Dasha
Horoscope	Moon sign, date	Daily, weekly and monthly guidance text
Panchang	Date, location	Tithi, Nakshatra, Yoga, Karana, sunrise, sunset
Matching	Two birth records	Guna Milan score out of 36 with summary

C. Data Flow

The user first enters name, date, exact time and birth city. The city is converted to latitude, longitude and time zone using a place database. The local time is converted to Universal Time. The ephemeris engine then returns the position of each planet, and the Kundli service converts

these into signs, houses and Nakshatras. The result is shown as a chart with a short explanation beside each item.

D. Database Design

A relational database with three main tables is sufficient. The Users table stores an anonymous ID and an optional e-mail. The Profiles table stores birth details and calculated chart data. The Content table stores the text of horoscopes and explanations in different languages. Calculated results can be cached, since the same birth data always gives the same chart.

E. Interface Design (API)

Each module is exposed through a small set of endpoints that accept and return JSON. Typical examples are POST /kundli for chart creation, GET /panchang for the daily almanac, GET /horoscope for sign-wise text and POST /match for compatibility. Every request is validated on the server: dates must be real, times must be within 24 hours, and place names must be found in the place database. Clear error messages tell the user exactly which field needs correction.

F. Security Design

All traffic uses HTTPS. Passwords, if accounts are enabled, are stored only as salted hashes. Birth data in the database can be encrypted at rest, and access to the database is limited to the application service. Rate limiting protects the API from misuse, and server logs avoid storing full birth details.

G. Scalability and Caching

Since a chart for fixed birth data never changes, results can be cached and reused. The daily Panchang is the same for all users in one city, so it is calculated once per day per location and served from cache. Horoscope text is prepared in advance and stored in the Content table, so it needs no heavy computation at request time. With these steps, the number of ephemeris calls stays low, and the application can serve many users on modest hardware.

V. IMPLEMENTATION

A. Technology Stack

The front end is built with HTML, CSS and JavaScript using a responsive layout so that it works on phones and desktops. The back end uses a REST-style web service, and the database can be PostgreSQL or MySQL. For astronomy, a standard ephemeris library is used rather than writing formulas from scratch, which reduces errors.

B. Chart Calculation

Planet positions from the ephemeris are tropical. They are converted to sidereal longitude by subtracting the ayanamsa A(t) for the date:

$$\lambda_s = \lambda_t - A(t) \qquad (1)$$

The Rashi number r and the Nakshatra number n are then found by simple division:

$$r = \lfloor \lambda_s / 30° \rfloor + 1 \quad (2)$$

$$n = \lfloor \lambda_s / 13.333° \rfloor + 1 \quad (3)$$

Here the Moon's longitude is used for the birth Nakshatra. The Vimshottari Dasha order is Ketu 7, Venus 20, Sun 6, Moon 10, Mars 7, Rahu 18, Jupiter 16, Saturn 19 and Mercury 17 years, totalling 120 years. The starting planet is decided by the birth Nakshatra, and the balance of the first period is calculated from how much of that Nakshatra the Moon has already crossed.

The same approach gives the Panchang. Using the sidereal longitudes of the Sun (λ_S) and the Moon (λ_M), the Tithi t and Yoga y are:

$$t = \lfloor (\lambda_M - \lambda_S) \bmod 360° / 12° \rfloor + 1 \quad (4)$$

$$y = \lfloor (\lambda_S + \lambda_M) \bmod 360° / 13.333° \rfloor + 1 \quad (5)$$

A Tithi therefore lasts as long as the Moon takes to gain 12 degrees on the Sun, and a Karana is half of a Tithi. Sunrise and sunset are computed for the user's latitude and longitude, because the Hindu day (Vara) starts at sunrise and not at midnight.

In summary, the Kundli service performs the following steps for every request:

1) validate input and convert the place name to coordinates and time zone;
2) convert local birth time to Universal Time;
3) call the ephemeris for the nine planets (Rahu and Ketu come from the lunar node);
4) apply the ayanamsa and compute Rashi, Nakshatra and house for each planet;
5) find the Lagna from sidereal time and latitude, then build the Dasha table;
6) save the result in the cache and return it to the interface.

C. Guna Milan

Guna Milan compares the Moon Nakshatra and Rashi of two people over eight factors (kootas). Table III lists the maximum points; their total is 36. A score of 18 or more is traditionally seen as acceptable, but the app shows the score only as guidance and explains each factor separately.

TABLE III KOOTAS AND MAXIMUM POINTS IN GUNA MILAN

Koota	Points	Koota	Points
Varna	1	Graha Maitri	5
Vashya	2	Gana	6
Tara	3	Bhakoot	7
Yoni	4	Nadi	8

As a simple illustration, the Nadi koota gives 8 points only when the two people belong to different Nadi groups. If both share the same Nadi, it gives zero. The app shows this rule next to the score so that the user understands why the total is high or low, instead of seeing only a number.

D. User Interface

Usability guidelines [8] were followed. The birth form has only four fields and gives an example for each. Difficult words, such as Lagna or Dasha, show a short tooltip in simple language. The result page uses large chart images, short paragraphs and a clear note that the content is for guidance and entertainment. Pages are kept light so that they load quickly on slow mobile networks.

E. Multilingual Content

Because many users are more comfortable in Hindi than in English, all labels and explanations are stored as language keys instead of fixed text. Planet names, signs and Nakshatras use their traditional names, with a short English meaning beside them. A new language can be added by supplying one more translation file, without changing any calculation code.

F. Worked Example

Consider a user born on a given date at 06:30 in Lucknow. The place database returns the latitude, longitude and the Indian Standard Time offset of +5:30. The app converts 06:30 IST to Universal Time (01:00 UT) and requests the planet positions for that instant. Suppose the ephemeris gives the Moon a tropical longitude of 100° and the ayanamsa is 24°. By (1) the sidereal longitude is 76°; by (2) the Rashi number is 3 (Gemini, or Mithuna); and by (3) the Nakshatra number is 6 (Ardra). The Dasha table then starts with Rahu, the lord of Ardra. The page shows these results with one line of explanation for each, for example, “Your Moon sign is Mithuna, which is linked in tradition with curiosity and communication.” The values here are chosen only to show the method and are not from a real person.

G. Deployment and Maintenance

The application can be deployed on a standard cloud server with a web server in front of the API. Source code is kept in a version-control system, and automatic tests run before every release so that a change in one module does not silently damage another. Ayanamsa values, time-zone rules and place data change over time, so these data files are reviewed periodically. User feedback is also checked regularly to find unclear text or confusing screens, and the content is edited accordingly.

VI. TESTING STRATEGY

Testing is planned on four levels, as summarised in Table IV. The most important is accuracy testing, where the app’s planet positions and Panchang values are compared with an independent reference ephemeris and with printed almanacs for sample dates. Quality attributes such as functional suitability, performance and usability follow the ISO/IEC 25010 model [9].

TABLE IV TEST PLAN

Level	What is tested	Pass criterion
Unit	Ayanamsa, Rashi, Nakshatra, Dasha functions	Matches hand-calculated values
Accuracy	Planet longitudes and Panchang for sample dates	Within small tolerance of reference
Usability	Form completion by first-time users	Users finish without help
Performance	Page and chart response time	Fast on mobile networks

Edge cases receive special attention: births near midnight, births near sign boundaries, unknown birth time, places in different time zones, and historical dates when daylight saving rules differed. If the birth time is unknown, the app warns the user that the Lagna and houses cannot be trusted and shows only Moon-based results.

A. Expected Outcomes

When the above tests are met, the application should produce charts that match standard software and printed almanacs, load quickly on ordinary phones, and be understood by users who have no background in astrology. Early feedback from a small group of students and family members can be collected through a short questionnaire on clarity, trust and ease of use. The feedback is then used to improve wording and layout before a public release.

B. Discussion

The design shows that the technical part of a Vedic astrology app is a well-defined computing problem: given a time and a place, planetary positions follow from astronomy, and the rest is rule-based mapping. The challenging part is presentation. A correct chart is of little value if the user cannot understand it. For this reason the project spends as much effort on language and layout as on calculation.

VII. PRIVACY, ETHICS AND LIMITATIONS

Birth details are personal data. The application collects only what is needed, uses encrypted connections, allows deletion of all stored profiles, and does not sell data to third parties. These practices are in line with the principles of India's Digital Personal Data Protection Act, 2023 [6].

Ethical care is equally important. Astrology is a traditional belief system and its predictions have not been proved by scientific testing. For this reason the app avoids fear-based language, never gives medical, legal or financial instructions, and does not claim certainty about marriage, health or death. Users are encouraged to treat the content as cultural guidance and to make major life decisions with proper professional advice.

The main limitations of the current design are: results depend on the correctness of the birth time entered by the user; different schools use different ayanamsa and house systems; and horoscope text is general, so it cannot match every person's situation.

VIII. FUTURE SCOPE

Several improvements are possible. Divisional charts (such as Navamsa), Sade Sati and Mangal Dosha analysis can be added. Push notifications can inform users about daily auspicious times. Voice input and more regional languages would help users who are not comfortable with typing English. Finally, a user-selectable setting for ayanamsa and house system would suit advanced users and professional astrologers.

IX. CONCLUSION

This paper presented the design and implementation plan of vedic-astrology-fortune.app, a modular web application that produces a Kundli, horoscope, Panchang and Guna Milan score from basic birth details. By using a trusted ephemeris, explaining every term in simple language and protecting user data, the system offers a practical alternative to cluttered astrology apps. The layered architecture keeps the code easy to maintain and extend. The work also reminds developers that such an app should be presented honestly, as cultural and entertainment content, and not as scientific prediction.

ACKNOWLEDGMENT

The authors thank their teachers and classmates for guidance and feedback during this project.

REFERENCES

[1] J. Meeus, Astronomical Algorithms, 2nd ed. Richmond, VA, USA: Willmann-Bell, 1998.
[2] B. V. Raman, How to Judge a Horoscope. New Delhi, India: Motilal Banarsidass, 1990.
[3] Astrodienst AG, “Swiss Ephemeris,” [Online]. Available: https://www.astro.com/swisseph/
[4] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2002.
[5] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, Univ. California, Irvine, CA, USA, 2000.
[6] Government of India, “The Digital Personal Data Protection Act, 2023,” New Delhi, India, 2023.
[7] Council of Scientific and Industrial Research, Report of the Calendar Reform Committee. New Delhi, India: CSIR, 1955.
[8] J. Nielsen, Usability Engineering. San Diego, CA, USA: Academic Press, 1993.
[9] Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models, ISO/IEC 25010, 2011.