**Detailed Project Report**

**1. Project Overview**

Astrology Web and Mobile App Project Overview

1. Project Goal

We are going to Develop a comprehensive AI-powered astrology application for web and mobile platforms, providing users with a wide range of astrological features and personalized insights.

1. Key Features

|  |  |  |
| --- | --- | --- |
| S. No. | Features | Description |
| 1 | Kundali | - Kundali based on personal details |
| 2 | Kundali match for marriage | - Romantic compatibility between two users for marriage |
| 3 | Personalized Horoscopes  Today, Tomorrow, Weekly, Monthly | - Daily, weekly, and monthly horoscopes  - Customized based on user's birth chart |
| 4 | Love Horoscope  Today, Tomorrow, Weekly, Monthly | - Romantic compatibility between two users  - Friendship and professional relationship insights |
| 5 | Subh Muhurat |  |
| 6 | Panchang  Today, Monthly | - Important astrological events and dates  - Personalized favourable/unfavourable days |
| 7 | Planet Position | Transit and Progression Forecasts  - Real-time planetary positions and their effects  - Future predictions based on current transits |

1. Technical Specifications

- Platforms: Web (responsive), iOS, and Android

- Backend: Node.js with Express.js (Production server shall be hosted on AWS, Test Server shall be hosted on Vercel)

- Frontend: Flutter for web, mobile app

- Database: MongoDB for user and astrological data

- Schema’s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Admin | User | Astrologer | Zodiac Sign | Subscription |
| {      firstName: {          type: String,          required: true      },      lastName: {          type: String,          required: true      },      email: {          type: String,          required: true,          unique: true      },      password: {          type: String,          required: true      },      mobileNumber: {          type: String,          required: true      },      role: {          type: String,          enum: ['super-admin', 'admin'],          required: true      },      permissions: {          type: [String],          required: true      },      lastLogin: {          type: Date      },      status: {          type: String,          enum: ['Active', 'Inactive', 'Suspended'],          default: 'Active'      }  },      {          timestamps: true,      } | {          email: {              type: String,              required: true,              unique: true          },          password: {              type: String,              required: true          },          firstName: {              type: String,              required: true,              // unique: true          },          lastName: {              type: String,              required: true          },          gender: {              type: String,              enum: ['Male', 'Female', 'Other'],              required: true          },          mobileNumber: {              type: String,              required: true          },          profileImage: {              type: String,              required: true          },          birthDate: {              type: Date,              required: true          },          birthTime: {              type: String,              validate: {                  validator: function(v) {                      return /\b([01]?[0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]\b/.test(v);                  },                  message: props => `${props.value} is not a valid time format!`              },              required: true          },          birthPlaceDetails: [{              city: String,              state: String,              country: String,              latitude: Number,              longitude: Number          }],          preferences: [{              timezone: String,              language: String          }]      },      {          timestamps: true,      } | {      firstName: {          type: String,          required: true      },      lastName: {          type: String,          required: true      },      emailAddress: {          type: String,          required: true,          unique: true      },      password: {          type: String,          required: true,      },      gender: {          type: String,          enum: ['Male', 'Female', 'Other'],          required: true      },      birthDate: {          type: Date,          required: true      },      languagesSpoken: {          type: [String],          required: true      },      yearsOfExperience: {          type: Number,          required: true      },      specializations: {          type: [String],          required: true      },      consultationFees: {          perMinute: {              type: Number,              required: true          },          perSession: {              type: Number,              required: true          }      },      availability: {          workingHours: {              type: String,              required: true          },          consultationModes: {              type: [String],              required: true          }      },      profileImage: {          type: String,          required: true      },      contactNumber: {          type: String          required: true      },      ratings: {          type: Number,          default: 0      }  },      {          timestamps: true,      } | {      signName: {          type: String,          required: true      },      signSymbol: {          type: String,          required: true      },      signElement: {          type: String,          required: true      },      signQuality: {          type: String,          required: true      },      signDescription: {          type: String,          required: true      }  },      {          timestamps: true,      } | {      userId: {          type: String,          required: true      },      plan: {          type: String,          enum: ['Basic', 'Premium'],          required: true      },      startDate: {          type: Date,          required: true      },      endDate: {          type: Date,          required: true      },      status: {          type: String,          enum: ['Active', 'Inactive', 'Expired'],          required: true      }  },      {          timestamps: true,      } |
| Planet | Kundali | Feedback |  |  |
| {      planetName: {          type: String,          required: true,          unique: true      },      planetSymbol: {          type: String,          required: true      },      planetDescription: {          type: String,          required: true      },  },      {          timestamps: true,      } |  | {      userId: {          type: String,          required: true      },      content: {          type: String,          required: true      },      rating: {          type: Number,          min: 1,          max: 5,          required: true      },  },      {          timestamps: true,      } |  |  |

APIs:

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Feature | Request Type | APIs |
| 1 | User | POST | /api/users/registerUser |
|  |  | POST | /api/users/login |
|  |  | GET | /api/users/userProfile/:id |
|  |  | GET | /api/users/getAllUsers |
|  |  | PUT | /api/users/updateUserById/:id |
|  |  | DELETE | /api/users/deleteUserById/:id |
| 2 | Astrologer | POST | /api/astrologers/registerAstrologer |
|  |  | POST | /api/astrologers/login |
|  |  | GET | /api/astrologers/astrologerProfile/:id |
|  |  | GET | /api/astrologers/getAllAstrologers |
|  |  | PUT | /api/astrologers/updateAstrologerById/:id |
|  |  | DELETE | /api/astrologers/deleteAstrologerById/:id |
| 3 | Planet | POST | /api/planets/registerPlanet |
|  |  | GET | /api/planets/getAllPlanetDetails |
|  |  | GET | /api/planets/getPlanetInfoById/:id |
|  |  | PUT | /api/planets/updatePlanetInfoById/:id |
|  |  | DELETE | /api/planets/deletePlanet/:id |
| 4 | Admin | POST | /api/admin/registerAdmin |
|  |  | POST | /api/admin/login |
|  |  | PUT | /api/admin/updateAdminById/:id |
|  |  | GET | /api/admin/adminProfile/:id |
|  |  | GET | /api/admin/getAllAdmins |
|  |  | DELETE | /api/admin/deleteAdminById/:id |
| 5 | Subscription | POST | /api/subscriptions/createSubscription |
|  |  | GET | /api/subscriptions/getSubscription/:id |
|  |  | GET | /api/subscriptions/getUserSubscriptions/:userId |
|  |  | PUT | /api/subscriptions/updateSubscription/:id |
|  |  | DELETE | /api/subscriptions/deleteSubscription/:id |
| 6 | Feedback | POST | /api/feedbacks/createFeedback |
|  |  | GET | /api/feedbacks/getAllFeedbacks |
|  |  | GET | /api/feedbacks/getFeedbackById/:id |
|  |  | GET | /api/feedbacks/getFeedbackByUserId/:UserId |
|  |  | PUT | /api/feedbacks/updateFeedback/:id |
|  |  | DELETE | /api/feedbacks/deleteFeedback/:id |
| 7 | Zodiac Sign | POST | /api/signs/createSign |
|  |  | GET | /api/signs/getAllSigns |
|  |  | GET | /api/signs/getSign/:id |
|  |  | PUT | /api/signs/updateSign/:id |
|  |  | DELETE | /api/signs/deleteSign/:id |
| Postman APIs Collections URL  <https://api.postman.com/collections/28862280-963d0010-0449-4c66-ae05-974cb139d390?access_key=PMAT-01J66P4F87694BS3D0BYQ6MSJ7> | | | |

* Workflow Diagrams

|  |
| --- |
| Get Kundali |
| Step- 1 User shall access the app home page |
| Step- 2 User shall click on the Login button at homepage |
| Step- 3 User shall be navigated to the Login Screen |
| Step- 4 User shall Login using their credentials. If user not already registered, he shall select the “create account” option and shall be navigated to the Sign-Up Page. |
| Step- 5 User will fill all the details and get signed up. |
| Step-6 User shall be navigated the Dashboard. |
| Step-7 User shall Click on the Get Kundali Button. |
| Step-8 User shall be navigated to the Get Kundali page. |
| Step-9 After all the details on Get Kundali page user shall get the Kundali. |

* UI Designs

<https://www.figma.com/design/jf0Gt8hwRjBEQnCABRVlqz/Astro?node-id=0-1&t=0ZV3nV5Dmf9fUefe-1>

1. Development Phases

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Details of Phase | Days / Weeks / Months | Status |
| 1 | Research and Planning | 1 week (7 days) | Done |
| 2 | Design and Prototyping | 1 weeks (7 days) | Done |
| 3 | Core Feature Development | 4 weeks (28 days) | In Progress |
| 4 | Testing and Quality Assurance | 2 day |  |
| 5 | Beta Launch and Feedback Collection | 1 day |  |
|  | Total required days | 6 weeks 2 days **OR**  45 Days |  |
|  | Remaining days | 3 weeks approx. |  |

This project aims to create a comprehensive, user-friendly, and accurate astrology application that caters to both beginners and advanced enthusiasts in the field of astrology.

**2. Plan of Action and Current Progress**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | Action | Assigned to Team Member | Status |
| 1 | Mongo DB Database shall be created that include the following schema’s   1. Admin 2. User 3. Astrologer 4. Zodiac Sign 5. Subscription 6. Planet 7. Kundali 8. Feedback | Alok | Completed |
| 2 | Mongo DB Test Database shall be created that include the following schema’s   1. Admin 2. User 3. Astrologer 4. Zodiac Sign 5. Subscription 6. Planet 7. Kundali 8. Feedback | Alok | In Progress |
| 3 | Node JS server shall be created with CRUD operation for the below mentioned schema’s   1. Admin 2. User 3. Astrologer 4. Zodiac Sign 5. Subscription 6. Planet 7. Kundali 8. Feedback | Alok | Completed |
| 4 | Production server shall be deployed to the Amazon Web Service. | Alok | In Progress |
| 5 | Test server shall be deployed to the Vercel platform. | Alok | Completed |
| 6 | JWT token implementation for secured login of various users | Alok | In Progress |
| 6 | Deploy the AI Astro Server to local host 5000 port | Alok | In Progress |
| 7 | UI Design of 9 screen shall be created | Shikhar | Completed |
| 8 | Project Report Compilation | Shikhar | In Progress |
| 9 | Eight Screen UI Design in flutter | Arun | In Progress |
| 10 | Business Logic for Login, Sign Up, Kundali Generation. | Arun | In Progress. |

**3. Timeline**

[Gantt chart attached]