# Project: Pariksha Mitra (परीक्षा मित्र)

**Description:**
Pariksha Mitra is a comprehensive digital examination platform for Marathi medium students from 5th to 10th standard. The platform aims to transform exam preparation through:

- **Personalized practice tests**
- **Detailed performance analytics**
- **Teacher-guided feedback tests**
- **Interactive learning experiences through games**

## 1. Role: UI/UX Designer

**Objective:**
Design a user-centric experience for a Marathi medium exam portal that is:

- Intuitive
- Engaging
- Culturally relevant
- Accessible to students aged 10-16

**Design Challenges:**

1. **User Interface:**
   - Design interface in Marathi language.
   - Use appealing color schemes for students.
   - Develop culturally relevant icons and visual metaphors for the Indian educational context.
2. **User Experience:**
   Design user flows for:

- Test selection.
- Practice test taking.
- Feedback test participation.
- Performance review.

3. **Accessibility:**
    - Ensure readability for young students.
    - Create responsive designs for various devices.
    - Optimize for low-bandwidth scenarios.

**Deliverables:**

- Figma/Adobe XD prototype
- Design system document
- User journey maps
- Accessibility compliance report
- Cultural relevance analysis

**Evaluation Criteria:**

- Cultural sensitivity
- User-friendliness
- Visual appeal
- Innovative navigation design
- Alignment with project goals

---

## 2. Role: MERN Full Stack Developer

**Responsibilities:**

1. **Project Setup:**
    - Initialize MERN project structure.
    - Configure environment variables.
    - Set up MongoDB connection.
2. **Backend Development:**
    - Implement user model.
    - Create authentication routes:

        - /api/auth/register
        - /api/auth/login
    - Implement password hashing.
    - Generate and validate JWT tokens.
    - Create middleware for role-based access.
3. **Frontend Development:**
    - Develop registration and login pages.
    - Implement protected routes and authentication state.
    - Display user information post-login.
4. **Deployment:**
    - Deploy on Render.
    - Configure environment settings.
    - Set up continuous integration.

**Evaluation Criteria:**

- Code quality
- Security implementations
- Functional authentication flow
- Responsive design
- Successful deployment

---

# 3. Role: Node.js Backend Developer

**Responsibilities:**

1. **API Design:**
    - Create Swagger documentation.
    - Implement RESTful endpoints:
        - POST /register
        - POST /login
        - GET /user-profile
2. **Database Integration:**
    - Design MongoDB schema and implement user model.

    - Establish database connection.

3. **Authentication Mechanisms:**
    - Password hashing using bcrypt.
    - Generate and validate JWT tokens.
    - Implement role-based access control.
4. **Security Implementations:**
    - Input validation and error handling.
    - Secure token management.

**Evaluation Criteria:**

- API design and Swagger documentation.
- Security implementations.
- Error handling.
- Database interaction.

**Recommended Tools:**

- Backend: Express.js
- Authentication: JSON Web Tokens (JWT)
- Password Hashing: bcrypt
- API Documentation: Swagger
- Hosting: Render
- Database: MongoDB Atlas

---

## 4. Role: React Front-End Developer

**Responsibilities:**

1. **UI/UX Design:**
    - Develop responsive registration form and login page.
    - Create dashboard layout.
2. **State Management:**
    - Implement authentication state.
    - Use dummy data for login simulation.
    - Create protected route components.

3. **Form Handling:**
    - Implement form validation.
    - Handle user interactions.
    - Display error messages.
4. **Dashboard Development:**
    - Display user information.
    - Implement mock login functionality.

**Evaluation Criteria:**

- UI/UX design and responsiveness.
- React best practices and component structure.
- Effective state management.

---

## 5. Role: Data Analyst

**Project: Data Architecture and Management for Pariksha Mitra**

**Objective:**
Design an efficient data strategy focusing on storage, retrieval, and utilization.

**Specific Tasks:**

1. **Data Modeling:**
    - Create schemas for:
        - Student profiles.
        - Chapter structure.
        - Exercise definitions.
        - Question bank.
        - Test results.
        - Performance analytics.
2. **Data Storage Strategy:**
    - Recommend database structures with efficient indexing.
    - Create data relationship diagrams.
    - Propose data privacy and security measures.
3. **Data Processing:**
    - Develop algorithms for:

        - Question randomization.
        - Performance calculation.
        - Analytics generation.

**Deliverables:**

- Data architecture document.
- MongoDB/NoSQL schema designs.
- Data flow diagrams.
- Performance optimization recommendations.

**Evaluation Criteria:**

- Depth of data modeling.
- Scalability of solutions.
- Clarity of documentation.
- Innovative problem-solving approach.

---

**Submission Requirements**

- GitHub repository
- Deployed application link
- README with setup instructions
- Postman/Swagger API documentation

This roadmap ensures the successful implementation of Pariksha Mitra, balancing user-centric design, robust backend systems, and scalable data strategies.

---

## 6. Game Developer

Develop any 2 games

| Category | Game | Educational Theme | Implementation Details (Unity/Godot) |
|---|---|---|---|
| **JavaScript Games** | Basic Platformer | Math Challenges | Players solve math puzzles to unlock doors or activate platforms (e.g., solve equations, find prime numbers). |
| | Snake Game | Number Sequences | Players collect numbers in a sequence (e.g., multiples of 3, Fibonacci) to grow the snake. |
| | Tic-Tac-Toe | Math Strategy | Use numbers to make sums equal a target value (e.g., 15) instead of traditional Xs and Os. |
| | Asteroids Shooter | Geometry | Destroy asteroids by solving geometry problems (e.g., angles, shapes). |
| | Memory Card Game | Math Equations | Match cards with equations and their solutions (e.g., 3x3 matches with 9). |
| **Game Engine Games** | Lantern Chase | Math Puzzles in Dark | Collect light orbs to solve puzzles (e.g., area, probability) while avoiding enemies. |
| | Musical Tiles | Math Rhythm Game | Hit tiles based on math problem answers in sync with the music (e.g., solve 2x2 to hit the "4" tile). |
| | Spider Web Builder | Coordinate Geometry | Draw webs by solving coordinate geometry problems or forming specific shapes. |
| | Time Warp Maze | Math and Time | Solve time-based equations (e.g., elapsed time, speed-distance) to manipulate the maze's time flow. |
| | Bridge the Gaps | Physics and Math | Build bridges by calculating forces, weights, or costs; solve algebra/physics equations for challenges. |