SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

CHESS ACADEMY

A PROJECT REPORT

Submitted by

S M ASHIKA

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY
<Font Size 16>

IN

ARTIFICIAL INTELLIGENCE AND DATA
SCIENCE

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)

JULY-AUGUST & 2024

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

BONAFIDE CERTIFICATE

Certified that this project report **CHESS ACADEMY** is the bonafide work of **S M ASHIKA** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
DR.S. VENKATA LAKSMI	<<Name>>
HEAD OF THE DEPARTMENT,	
AIDS	**SUPERVISOR**

Submitted for the Project viva-voce examination held on______________

INTERNAL EXAMINER	EXTERNAL EXAMINER

TABLE CONTENTS

Project Overview:

The Chess Academy website is designed to provide a seamless online experience for chess enthusiasts, instructors, and students. The site integrates a comprehensive set of features, including landing pages, user authentication, course management, classrooms, and tournament management. The backend of the website is powered by Django, while the frontend is built using React. The combination ensures a dynamic and responsive user interface while maintaining secure and efficient data handling.

1. Website Structure

2.1 Landing Page

Purpose: The landing page is the gateway to the Chess Academy. It aims to attract users with a visually appealing design and clear call-to-action buttons.

Features:

- Hero section with academy branding.
- Brief introduction to the academy.
- Call-to-action buttons like "Join Now" or "Request Call Back".
- Testimonials or highlights of academy achievements.

Design:

- Responsive design with a fluid grid layout.
- Eye-catching imagery, including chess-related visuals.
- Simple, clean navigation.

2.2 About Page

Purpose: To provide detailed information about the Chess Academy, its mission, and its team.

Features:

- History and mission statement of the academy.
- Profiles of the instructors and founders.
- Academy milestones and achievements.

Design:

- Structured sections with images and text.
- Scrollable timeline of achievements.

Backend Integration:

- Instructor profiles stored in Django models and fetched via an API.
- Content management through Django admin.

2.3 Contact Page

Purpose: To enable users to get in touch with the academy.

Features:

- Contact form for inquiries.
- Google Maps integration for location display.
- Contact information including phone, email, and address.

Design:

- Simple, user-friendly form design.
- Responsive map display.

Backend Integration:

- Submissions from the contact form are stored in the Django database.
- Admin notifications for new inquiries.

2.4 Home Page

Purpose: Serves as the dashboard for logged-in users, providing quick access to courses, classrooms, and tournaments.

Features:

- Quick links to different sections of the website.
- Summary of user activity, such as courses enrolled and tournaments participated in.

Design:

- Dashboard-style layout with cards for each feature.
- Personalized greeting based on user data.

Backend Integration:

- User-specific data fetched from Django, including courses and tournaments.
- Secure access with JWT token authorization.

2.5 Add Courses Page

Purpose: Allows instructors to add new courses to the academy's offerings.

Features:

- Form to enter course details, including title, description, and price.
- Upload feature for course materials and images.

Design:

- Clean, form-based interface.
- Validation to ensure all required fields are filled.

Backend Integration:

- Course data is saved in the Django database.
- File uploads are handled by Django's file storage system.

2.6 Classrooms for Instructor

Purpose: Provides instructors with tools to manage their classes, including assignments and student interactions.

Features:

- Dashboard to view all ongoing courses.
- Ability to add and grade assignments.
- Interaction section for announcements and discussions.

Design:

- Structured, tabbed layout for easy navigation.
- Integration of rich text editors for assignments and announcements.

Backend Integration:

- Instructor data and course materials are managed through Django.
- Real-time updates using Django Channels (for interactions).

2.7 Classrooms for Student

Purpose: A dedicated space for students to access their enrolled courses, submit assignments, and interact with instructors.

Features:

- List of enrolled courses with progress tracking.
- Assignment submission portal.
- Discussion board for course-related queries.

Design:

- Interactive, student-friendly design.
- Visual indicators of progress.

Backend Integration:

- Data synchronization with Django backend to fetch student assignments and submissions.
- Real-time feedback and grading updates.

2.8 Tournament Conduction Page

Purpose: Allows users to organize and manage chess tournaments.

Features:

- Form to create a new tournament with details like date, participants, and format.
- Management dashboard for ongoing tournaments.

Design:

- Step-by-step wizard for tournament creation.
- Overview page for tournament statistics and matchups.

Backend Integration:

- Tournament data is stored in Django models.
- API endpoints for fetching and updating tournament details.

2.9 Tournament Joining Page

Purpose: Enables users to browse and join tournaments.

Features:

- List of available tournaments with search and filter options.
- Option to join a tournament directly from the list.

Design:

- Card-based layout for easy browsing.
- Detailed view of each tournament with a "Join Now" button.

Backend Integration:

- Data retrieval from Django to display available tournaments.
- Enrollment process handled via API calls.

2.10 Login & Signup Pages

Purpose: Handles user authentication for accessing the website's features.

Features:

- Separate forms for login and signup.
- Validation for input fields.
- Password recovery option.

Design:

- Minimalist design with focus on usability.
- Error handling for invalid credentials.

Backend Integration:

- User authentication managed via Django's built-in auth system with JWT tokens for React.
- Signup process includes validation and email confirmation.

2. Backend Implementation (Django)

3.1 Models

- **User**: Custom user model to include additional fields like `phone` and `address`.
- **Course**: Stores course information, including title, description, and instructor.
- **Tournament**: Manages tournament details, including participants and schedule.
- **Classroom**: Separate models for instructor and student interactions.
- **Contact**: Stores contact form submissions.

3.2 Views

- **APIs**: Django Rest Framework (DRF) is used to create API views for all interactions with the frontend.
- **Authentication**: JWT-based authentication to secure access to protected resources.

3.3 Serializers

- Each model has a corresponding serializer to control the data exchanged between the frontend and backend.

3.4Admin Interface

- Admins can manage all aspects of the website, including user data, courses, tournaments, and more through Django's admin interface.

3. Frontend Implementation (React)

4.1 Routing

- React Router is used to manage navigation between pages. Each component (e.g., `HomePage`, `LoginPage`, `AddCoursesPage`) is associated with a route.

4.2 State Management

- Global state is managed using React Context or Redux, ensuring consistent data flow across components.

4.3 API Integration

- Axios is used for making API requests to the Django backend. JWT tokens are attached to headers for authenticated requests.

4.4 Responsive Design

- CSS Flexbox and Grid are used to create a responsive layout, ensuring the website is accessible on all devices.

4.5 Animations

- Framer Motion is used to add animations for smoother transitions and better user experience.

4. Deployment

- **Frontend**: The React app is deployed using services like Vercel or Netlify.

- **Backend**: The Django backend is deployed on a server with an Apache or Nginx configuration, using services like Heroku, AWS, or DigitalOcean.

- **Database**: The MySQL database is hosted on a managed service like AWS RDS or hosted directly on the server.

- **Static and Media Files**: Managed using Django's `collectstatic` for static files and an external service like AWS S3 for media files.

5. Security Considerations

- **Authentication**: JWT tokens with secure handling to prevent unauthorized access

  .

- **CSRF Protection**: Implemented for form submissions.

- **Data Validation**: Extensive validation on both frontend and backend to prevent malicious input.

6. Conclusion

The Chess Academy website is a robust platform that leverages modern web technologies to deliver a comprehensive, user-friendly experience for both students and instructors. The seamless integration between the React frontend and Django backend ensures that the application is scalable, secure, and easy to maintain.

Future enhancements could include the integration of more advanced features like AI-based chess tutors, live tournaments with real-time commentary, and mobile app development to reach a broader audience.