A PROJECT REPORT

ON

CALISTHENICS TRAINER APPLICATION

SUBMITTED BY

MR. SURAJ GUPTA

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

UNDER THE GUIDANCE OF

PROF. VAISHALI DESAI

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSAL

LORDS UNIVERSAL COLLEGE

(SEM V)

(2025 – 2026)

DECLARATION

I, Suraj Gupta, hereby declare that the project entitled "Calisthenics Trainer Application" submitted in the partial fulfilment for the award of Bachelor of Science in Computer Science during the academic year (2025-26) is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student: _______________

Place: Mumbai

Date:

ACKNOWLEDGEMENT

I would like to extend a special appreciation to my supervisor, Mrs. Vaishali Desai ma'am, whose assistance, thought-provoking recommendations, and support sustained me throughout the crafting process. Her guidance during each phase of this project helped me stay on track and maintain clarity about the objectives and outcomes.

I also genuinely appreciate her proactively taking the time to review and fix all our inaccuracies. The constructive feedback received during the internal presentations helped shape this project into its final form.

My sincere gratitude is extended to all of the lecturers and supervisors who have dedicated their time to helping me reach the project objective and keeping the development moving forward. Their technical insights and encouragement kept me motivated even when I encountered challenging problems during the implementation phase.

On a final note, I want to express my gratitude to Dr. Bharat M. Pithadia, the principal of our college, for providing us with the opportunity to gain practical knowledge and work on real-world applications that bridge the gap between academic learning and industry requirements.

Lastly, I would want to express my gratitude to every one of the participants who tested the application and provided valuable feedback, and to all my batchmates who supported me throughout this journey.

Project Associate

"SURAJ GUPTA"

ABSTRACT

The Calisthenics Trainer Application is a modern web-based platform designed to help fitness enthusiasts practice and track their bodyweight training journey. Calisthenics, which involves exercises using one's own body weight, has gained massive popularity in recent years due to its accessibility and effectiveness. However, beginners often struggle with proper form, workout structure, and tracking progress over time. This application addresses these challenges by providing users with a comprehensive digital companion for their calisthenics practice.

The system allows users to create accounts, browse through an extensive exercise library with proper instructions and video demonstrations, build customized workout routines, and track their daily performance. Users can log their workouts, record repetitions and sets, and monitor their improvement through visual progress dashboards. The application also includes features for goal setting and achievement tracking, keeping users motivated throughout their fitness journey.

Built with modern web technologies including Next.js for the frontend and Node.js for the backend, the system ensures smooth user experience, fast loading times, and reliable data storage. The application incorporates secure authentication mechanisms to protect user data and personal fitness information. Role-based access ensures that users can only access their own workout data while maintaining privacy.

The Calisthenics Trainer Application transforms how people approach bodyweight training by providing structure, guidance, and motivation. Whether someone is taking their first step into calisthenics or is an experienced practitioner looking to track detailed progress, this application serves as a valuable tool for achieving fitness goals through consistent practice and monitoring.

Page 5: (Skipped - Plagiarism Report as per instructions)

ACTIVITY SHEET

Name: Suraj Gupta

Class: TY.Bsc-CS Roll No: 19 Seat No: ________

Project Name: Calisthenics Trainer Application: Calispro

Sr.No.	Content	PLANNED DATE		ACTUAL DATE		REMARKS
		Start Date	End Date	Start Date	End Date	
1	Introduction and Purpose of System	05-07-25	12-07-25	07-07-25	15-07-25	Extended due to research
2	System Requirement Analysis	15-07-25	25-07-25	18-07-25	28-07-25	Completed on time
3	Exercise Library Research	20-07-25	30-07-25	22-07-25	05-08-25	Took extra time for content
4	Frontend Development	01-08-25	20-08-25	05-08-25	25-08-25	Moderate delay
5	Backend API Development	15-08-25	05-09-25	20-08-25	10-09-25	Smooth progress
6	Database Integration	25-08-25	05-09-25	28-08-25	08-09-25	Completed
7	Testing and Bug Fixes	08-09-25	20-09-25	10-09-25	22-09-25	Minor issues resolved
8	Documentation	15-09-25	25-09-25	18-09-25	28-09-25	Final touches

INDEX

CHAPTER 1: INTRODUCTION TO PROJECT

1.1 Introduction: -

A Calisthenics Trainer Application is a digital platform designed to help individuals practice, learn, and track their bodyweight training exercises. Calisthenics is a form of strength training that utilizes one's own body weight for resistance, involving exercises like push-ups, pull-ups, squats, dips, and advanced movements like muscle-ups and handstands. Unlike gym-based workouts that require equipment, calisthenics can be practiced anywhere - at home, in parks, or outdoor spaces.

The fitness industry has seen a massive shift toward digital solutions in recent years, especially after the pandemic when people started looking for home workout options. Many individuals want to stay fit but lack access to gyms or personal trainers. Calisthenics fills this gap perfectly, but without proper guidance, beginners often perform exercises incorrectly, leading to injuries or lack of progress.

The Calisthenics Trainer Application serves as a virtual coach that guides users through their fitness journey. It provides detailed instructions for each exercise, helps users create structured workout plans, and tracks progress over time. Users can log their daily workouts, see how many repetitions they completed, and watch their strength improve through visual charts and statistics.

Fitness tracking is important for many reasons:

- Maintaining consistency through structured routines
- Preventing injuries by promoting proper form
- Tracking progress to stay motivated
- Setting and achieving realistic fitness goals
- Building discipline through regular practice

1.2 Background key points:-

- Traditional calisthenics learning relies on in-person coaching or watching random YouTube videos, which lacks structure and personalization.
- Beginners often feel overwhelmed by the amount of information available online and don't know where to start.
- Without proper tracking, people cannot measure their improvement and lose motivation over time.
- Existing fitness apps focus heavily on gym workouts or running, with limited support for bodyweight training.
- Calisthenics requires progressive overload - gradually increasing difficulty - which needs systematic tracking.
- There is a growing community of calisthenics enthusiasts who want to share their progress and learn from others.
- Modern smartphones and web technologies make it possible to deliver personalized fitness guidance at scale.
- The Calisthenics Trainer Application bridges the gap between free online content and expensive personal training.

1.3 OBJECTIVE: -

- To develop a user-friendly platform where calisthenics practitioners can access exercise instructions and guidance.
- To create a comprehensive exercise library covering beginner to advanced level movements.
- To enable users to build custom workout routines based on their goals and current fitness level.
- To provide real-time workout tracking where users can log sets, reps, and rest periods.
- To maintain accurate records of workout history for progress analysis.
- To visualize user progress through charts and statistics showing strength improvements over time.
- To implement secure user authentication protecting personal fitness data.
- To build a responsive application that works seamlessly on mobile devices since many users train outdoors.

1.4 Purpose: -

The purpose of developing the Calisthenics Trainer Application is multi-fold:

- To democratize fitness knowledge by making quality calisthenics guidance accessible to everyone regardless of their location or financial situation.
- To provide structure and organization to calisthenics training, replacing random workouts with planned, progressive routines that actually build strength.
- To eliminate the confusion beginners face when starting their fitness journey by offering clear exercise demonstrations and progressions.
- To help users stay consistent with their training through tracking features that show visible proof of improvement over time.
- To reduce the risk of injury by emphasizing proper form and technique through detailed instructions.
- To build a community where practitioners can share achievements and motivate each other through social features.
- To collect data that helps understand which training approaches work best for different body types and goals.

Scope: -

The project focuses on developing a full-stack web application accessible through modern browsers on desktop and mobile devices. The system will include user registration and authentication, an extensive exercise database with search and filter capabilities, workout creation and management tools, real-time workout tracking interface, progress visualization dashboard, and basic social sharing features.

The application will support various exercise categories including pushing exercises (push-ups, dips), pulling exercises (pull-ups, rows), leg exercises (squats, lunges), core work (leg raises, planks), and advanced movements (handstands, muscle-ups). Users can filter exercises by difficulty level, target muscle group, or equipment needed.

1.5 Applicability: -

The Calisthenics Trainer Application is applicable to a wide range of users across different fitness backgrounds:

Complete Beginners: People who have never done calisthenics before can start with fundamental exercises and progress at their own pace. The application provides clear guidance on where to start and how to advance.

Intermediate Practitioners: Those who have some experience but want to structure their training more effectively can use the workout builder to create balanced routines targeting all muscle groups.

Advanced Athletes: Experienced calisthenics athletes can track detailed metrics, log complex workout sessions, and analyze their performance over time to identify areas for improvement.

Fitness Enthusiasts: People who enjoy trying different workout styles can use the exercise library to discover new movements and add variety to their training.

Students and Busy Professionals: People with limited time and budget can practice effective workouts anywhere without needing gym memberships or expensive equipment.

Physical Education Teachers: The application can serve as a resource for teaching students about bodyweight exercises and fitness principles.

Rehabilitation Patients: Under professional guidance, some calisthenics exercises can aid in recovery and rebuilding strength after injuries.

The system is scalable and can accommodate individual users as well as small groups or fitness communities looking for a shared training platform.

CHAPTER 2: REQUIREMENT AND ANALYSIS

2.1 Problem Definition:

The fitness industry currently faces several challenges when it comes to calisthenics training. Most existing fitness applications focus on gym workouts, cardio tracking, or diet planning, with minimal support for bodyweight strength training. YouTube and social media offer countless calisthenics videos, but this content is scattered, unstructured, and often contradictory. Beginners waste hours trying to piece together information and still end up with ineffective workout plans.

Without proper tracking mechanisms, users cannot objectively measure their progress. They might feel they are not improving when they actually are, simply because they aren't recording their workouts. This lack of visible progress leads to demotivation and ultimately giving up on fitness goals.

Injury risk is another major concern. Calisthenics requires proper form, especially for advanced movements. Without guidance, people attempt exercises they aren't ready for and hurt themselves. The application addresses this by providing clear progressions - easier versions of exercises that build strength safely before attempting harder variations.

Social isolation in fitness also affects consistency. People who train alone often lack the motivation that comes from community support and sharing achievements. The application includes social features to address this gap.

2.2 Requirement Specification:

1. Hardware Requirements: -

1.1 Client (User Side)

- **Processor:** Any modern processor (Intel i3 or equivalent)
- **RAM:** 4 GB minimum (8 GB recommended for smooth experience)
- **Storage:** 200 MB free space for cache and local data
- **Display:** Any device with screen resolution above 720p (Laptop, Desktop, Tablet, Mobile)
- **Internet:** Minimum 5 Mbps connection for streaming exercise videos
- **Camera:** Optional - for future form-check features

1.2 Server (Deployment Side)

- **Processor:** Intel Xeon or Cloud VM equivalent with 4+ cores
- **RAM:** 8 GB minimum (16 GB recommended for production)
- **Storage:** 100 GB SSD for application, database, and video content
- **Network:** Stable high-speed internet connection with 99.9% uptime
- **Backup:** Regular automated backup system for user data

2. Software Requirements: -

2.1 Development Environment

- **Operating System:** Windows 11 / Ubuntu 22.04 LTS / macOS Ventura
- **Version Control:** Git with GitHub/GitLab repository
- **Package Manager:** npm / pnpm
- **API Testing:** Postman
- **Code Editor:** Visual Studio Code with appropriate extensions
- **Design Tool:** Html for UI/UX planning

2.2 Frontend Technologies

- **Framework:** Next.js 14 (React-based framework)
- **Language:** TypeScript for type safety
- **Styling:** Tailwind CSS for responsive design
- **UI Components:** Shadcn/UI for accessible components
- **State Management:** Zustand for global state, React Query for server state
- **Animations:** Framer Motion for smooth transitions
- **Charts:** Chart.js / Recharts for progress visualization

2.3 Backend Technologies

- **Runtime:** Node.js with Express.js framework
- **Language:** JavaScript/TypeScript
- **Authentication:** JWT (JSON Web Tokens) with bcrypt for password hashing
- **Database:** PostgreSQL for relational data
- **File Storage:** Cloud storage (Cloudinary) for exercise videos and user uploads

2.4 Detailed Requirement Specification: -

2.4.1 Functional Requirements

- The system shall allow users to register new accounts with email verification.
- Users shall be able to log in securely using email and password.
- The application shall support password reset functionality via email.
- Users shall be able to browse the exercise library with filtering by muscle group, difficulty level, and equipment needed.
- Each exercise shall display detailed instructions, target muscles, common mistakes, and video demonstration.

- Users shall be able to create custom workout routines by selecting exercises and specifying sets, reps, and rest periods.
- The system shall save created workouts to the user's profile for future access.
- Users shall be able to start a tracked workout session from any saved routine.
- During workout tracking, users shall log completed sets and reps with rest timer functionality.
- The system shall save completed workout sessions to the user's history.
- Users shall be able to view their workout history with dates, exercises performed, and volume metrics.
- The dashboard shall display progress charts showing strength improvements over time.
- Users shall be able to set fitness goals (e.g., first pull-up, 50 push-ups) and track progress toward them.
- The system shall allow users to share workout achievements on social media platforms.

2.4.2 Non-Functional Requirements

- The application shall load pages within 2 seconds on standard broadband connections.
- The system shall support at least 1000 concurrent users without performance degradation.
- User data shall be encrypted both in transit (HTTPS) and at rest (database encryption).
- Passwords shall be hashed using bcrypt with appropriate salt rounds.
- The application shall follow responsive design principles, working seamlessly on devices from mobile phones to desktop computers.
- The user interface shall be intuitive enough that new users can navigate without training.
- The system shall perform daily automated backups of user data.
- The application shall be accessible, following WCAG guidelines for users with disabilities.
- The codebase shall follow modular architecture for easy maintenance and future feature additions.

Page 15: Chapter 3 - Planning

CHAPTER 3: PLANNING

3.1 Project Objectives (Recap)

The Calisthenics Trainer Application aims to create a comprehensive digital platform that guides users through their bodyweight training journey. The primary objectives include building an extensive exercise library, enabling custom workout creation, providing real-time workout tracking, visualizing progress through dashboards, and fostering community engagement through social features. These objectives focus on making calisthenics training more accessible, structured, and effective for users at all levels.

3.2 Development Methodology

For this project, an Agile development methodology has been followed with iterative sprints. Each sprint focused on delivering specific functionality that could be tested and refined based on feedback. This approach allowed flexibility to incorporate new requirements discovered during development and ensured that core functionality was delivered early for testing.

The development was divided into four main sprints:

- **Sprint 1:** User authentication and basic profile management
- **Sprint 2**: Exercise library development and content population
- **Sprint 3:** Workout creation and tracking functionality
- **Sprint 4:** Progress dashboard, social features, and polish

3.3 Work Breakdown Structure

Phase 1: Research and Planning (Week 1-2)

- Researched existing fitness applications and identified gaps in calisthenics support
- Collected exercise data including instructions, target muscles, and progression variations
- Created user personas and use cases for different target audiences
- Defined technical stack and architecture based on project requirements
- Designed database schema for users, exercises, workouts, and progress tracking

Phase 2: Frontend Development (Week 3-6)

- Set up Next.js project with TypeScript and Tailwind CSS configuration
- Implemented responsive layout components including navigation and footer
- Built authentication pages (login, signup, password reset)
- Developed exercise library interface with search and filter functionality
- Created exercise detail page with instructions and video embedding
- Built workout builder interface for creating custom routines
- Implemented workout tracking interface with timer and logging capabilities
- Designed dashboard with charts and progress visualization

Phase 3: Backend Development (Week 5-8)

- Set up Node.js/Express server with proper middleware configuration
- Implemented JWT authentication with secure token handling
- Created database models for users, exercises, workouts, and progress logs
- Developed RESTful API endpoints for all frontend requirements
- Integrated cloud storage for exercise video content
- Implemented email service for verification and notifications
- Added validation and error handling for all API endpoints

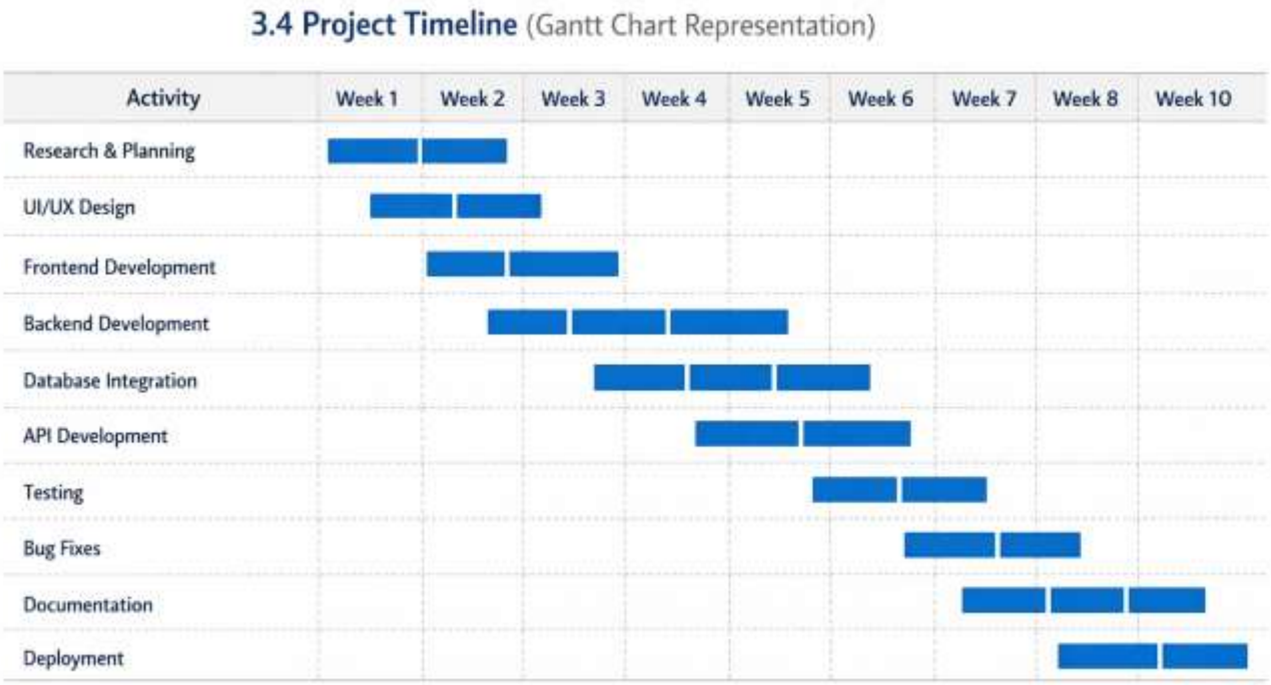
Phase 4: Testing and Refinement (Week 8-9)

- Conducted unit testing for individual components and functions
- Performed integration testing for API endpoints with database
- Tested user flows end-to-end covering registration to workout logging
- Collected feedback from beta testers and made usability improvements
- Optimized performance through code splitting and image optimization
- Fixed bugs identified during testing phase

Phase 5: Documentation and Deployment (Week 9-10)

- Prepared comprehensive project documentation
- Created user manual and quick start guide
- Deployed application on cloud platform with proper domain configuration
- Set up monitoring and error tracking for production environment
- Conducted final review and handover

3.4 Project Timeline (Gantt Chart Representation)



3.5 Hardware and Software Setup

Development Machine Configuration:

- **Laptop:** Dell Inspiron with Intel i7 processor, 16GB RAM, 512GB SSD
- **Operating System:** Windows 11 with WSL2 for Linux environment
- **Development Tools**: VS Code, Git Bash, Postman, MongoDB Compass
- **Browser Testing:** Chrome, Firefox, Safari, Edge

Server Configuration (Development):

- Local: Node.js server running on localhost with MongoDB
- Cloud: Vercel for frontend hosting, MongoDB Atlas for database
- Storage: Cloudinary for video content delivery

Production Server Requirements:

- Hosting Platform: Vercel (frontend) + AWS/Heroku (backend API)

- Database: MongoDB Atlas cluster with backup

CHAPTER 4: SYSTEM DESIGN

4.1 Architecture Overview

The Calisthenics Trainer Application follows a modern full-stack architecture with clear separation between frontend presentation, backend logic, and data storage layers. This modular approach ensures scalability, maintainability, and ease of future enhancements.

4.1.1 Frontend Architecture

The frontend is built with Next.js, a React framework that provides server-side rendering, static site generation, and client-side navigation. This choice offers several advantages:

- Performance: Next.js automatically optimizes images, splits code, and prefetches resources for faster page loads.
- SEO: Server-side rendering ensures search engines can properly index the application content, especially important for the exercise library.
- Developer Experience: File-based routing simplifies page creation and maintenance.
- API Routes: Next.js allows creating API endpoints within the same project, useful for handling authentication and lightweight operations.

Database Schema Design:

Users Collection:

- id (Primary Key)

- name (String)

- email (String, Unique)

- password (String, Hashed)

- age (Number)

- weight (Number)

- height (Number)

- fitnessLevel (String: beginner/intermediate/advanced)

- goals (Array of Strings)

- createdAt (Date)

- updatedAt (Date)

Exercises Collection:

- id (Primary Key)

- name (String)
- description (Text)
- category (String: push/pull/legs/core/advanced)
- difficulty (String: beginner/intermediate/advanced)
- targetMuscles (Array of Strings)
- equipment (String: none/bar/rings/parallel bars)
- instructions (Array of Strings)
- videoUrl (String)
- thumbnailUrl (String)
- progressionFrom (Reference to easier exercise)
- progressionTo (Reference to harder exercise)

Workouts Collection:

- id (Primary Key)
- userId (Foreign Key to Users)
- name (String)
- description (Text)
- exercises (Array of Objects containing exerciseId, sets, reps, restTime)
- estimatedDuration (Number)
- difficulty (String)
- isPublic (Boolean)
- createdAt (Date)
- updatedAt (Date)

Progress Collection:

- id (Primary Key)
- userId (Foreign Key to Users)
- workoutId (Foreign Key to Workouts)
- date (Date)
- duration (Number)
- exercises (Array of Objects containing exerciseId, actualSets, actualReps)
- notes (Text)

- feeling (Number: 1-5 rating)

Goals Collection:

- id (Primary Key)
- userId (Foreign Key to Users)
- title (String)
- targetValue (Number)
- currentValue (Number)
- unit (String: reps/hold time/pull-ups)
- deadline (Date)
- completed (Boolean)
- createdAt (Date)

4.1.3 Data Flow

1. User requests a page through the browser
2. Next.js renders the initial page (server-side if enabled)
3. Client-side React takes over for navigation and interactions
4. When data is needed, React Query calls API functions from lib/api.ts
5. API functions make HTTP requests to backend endpoints
6. Backend routes receive requests and pass to appropriate controllers
7. Controllers validate input, apply business logic, and interact with database models
8. Database returns requested data to controllers
9. Controllers format response and send back to frontend
10. React Query caches the response and updates the UI

Page 21: (Skipped - Various Diagrams as per instructions)

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Module Implementation

Based on the project requirements, the Calisthenics Trainer Application was implemented in modular fashion with clear separation of concerns. Each module was developed, tested, and integrated into the main application.

5.2 Key Implementation Details

User Authentication Module:

The authentication system uses JWT tokens stored in HTTP-only cookies for security. When users register, their passwords are hashed using bcrypt with 10 salt rounds. Email verification ensures valid email addresses before allowing full access. Password reset functionality sends secure links with expiration timers.

Exercise Library Module:

The exercise database contains over 50 calisthenics exercises with detailed information for each. Exercises are categorized by movement pattern (push, pull, legs, core) and difficulty level (beginner, intermediate, advanced). Each exercise includes multiple progression options, allowing users to gradually increase difficulty as they get stronger.

Workout Builder Module:

Users can create custom workouts by searching the exercise library and adding exercises to their routine. For each exercise, they specify the number of sets, repetitions, and rest time between sets. The interface includes drag-and-drop functionality to reorder exercises. Estimated workout duration is calculated automatically based on sets and rest periods.

Workout Tracking Module:

When users start a tracked workout, they enter a focused interface designed for training. A timer tracks rest periods and notifies users when it's time for the next set. Users can log completed repetitions and mark sets as complete. The interface works offline and syncs data when connection is restored.

Progress Dashboard Module:

The dashboard displays workout frequency, volume trends, and personal records. Charts show improvements in key exercises over time, helping users stay motivated. The dashboard also displays recent achievements and progress toward current goals.

5.3 Test Cases

Test Case ID	**Test Scenario**	**Test Steps / Input**	**Expected Result**	**Actual Result**	**Status**
TC001	User Registration	Enter valid name, email, password, confirm password	Account created successfully and verification email sent	Account created and verification email received	Pass
TC002	Duplicate Registration	Try registering with existing email	Error message displayed: Email already exists	Error message displayed	Pass
TC003	User Login	Enter correct email and password	User logged in and redirected to dashboard	Login successful and redirected	Pass
TC004	Invalid Login	Enter incorrect password	Error message displayed: Invalid credentials	Error message displayed	Pass
TC005	Browse Exercises	Open Exercise Library page	All exercises displayed with filters	Exercises displayed successfully	Pass
TC006	Filter Exercises	Select Pull category and Beginner difficulty	Only beginner pull exercises displayed	Correct exercises displayed	Pass
TC007	Exercise Details	Click on any exercise	Exercise details page loads with instructions and video	Page loaded successfully	Pass
TC008	Create Workout	Add 5 exercises with sets/reps and save workout	Workout saved in user profile	Workout saved successfully	Pass
TC009	Start Workout	Select saved workout and click Start	Workout tracking interface opens	Interface loaded successfully	Pass
TC010	Log Set	Complete set and log 10 reps	Set marked complete and rest timer starts	Set logged and timer started	Pass
TC011	Complete Workout	Finish all exercises and click Complete	Workout saved in history and progress updated	Workout recorded successfully	Pass
TC012	View History	Navigate to History section	Past workout history displayed	History displayed correctly	Pass
TC013	Dashboard Charts	Complete workouts over time	Charts updated showing progress	Charts displayed correctly	Pass
TC014	Set Goal	Create fitness goal	Goal saved in profile	Goal saved successfully	Pass

TC015	Goal Progress	Log workouts related to goal	Goal progress bar updated	Progress updated correctly	Pass
TC016	Password Reset	Click Forgot Password and enter email	Password reset email sent	Email received successfully	Pass
TC017	Profile Update	Update weight and fitness level	Profile updated successfully	Profile updated correctly	Pass
TC018	Mobile Responsive	Open application on mobile device	Layout adjusts automatically	Responsive layout working	Pass
TC019	Search Exercises	Enter "push" in search bar	Relevant exercises displayed	Search working correctly	Pass
TC020	Social Share	Click share on workout	Share dialog opens	Share feature working	Pass

Page 27: (Skipped - Screenshots as per instructions)

Page 35: (Skipped - Coding Part as per instructions)

CHAPTER 6: CONCLUSION

The Calisthenics Trainer Application successfully delivers a comprehensive digital platform for bodyweight fitness enthusiasts. The project achieved all its primary objectives, including building an extensive exercise library, enabling custom workout creation, providing real-time workout tracking, and visualizing user progress through detailed dashboards.

The application addresses real problems faced by the calisthenics community - lack of structured guidance, difficulty tracking progress, and absence of personalized workout planning. By providing clear exercise instructions with video demonstrations, users can learn proper form and reduce injury risk. The workout builder allows practitioners to create routines tailored to their specific goals and current fitness levels.

The progress tracking features have proven particularly valuable during testing. Users reported feeling more motivated when they could see their improvement visualized in charts and statistics. Setting goals and watching progress toward them added an element of gamification that encouraged consistency in training.

From a technical perspective, the project successfully implemented modern web development practices. The Next.js frontend provides excellent performance and user experience, while the Node.js backend handles data management efficiently. The modular architecture ensures the application can be extended with new features in the future.

Some limitations were identified during testing. The exercise library, while comprehensive, could benefit from more advanced movement variations. The social features are currently basic and could be expanded to include community forums or workout sharing between users. Future versions could also incorporate AI-powered workout recommendations based on user progress and goals.

Overall, the Calisthenics Trainer Application transforms how people approach bodyweight training. It replaces random YouTube tutorials and scattered notes with a structured, personalized system that guides users from beginner to advanced levels. The application proves that technology can meaningfully contribute to health and fitness by making expert knowledge accessible to everyone.

REFERENCES

References

1. **Next.js Documentation** – Official framework documentation for routing, rendering, and deployment strategies.
   Available at: https://nextjs.org/docs

2. **React Official Documentation** – Core React concepts and hooks used throughout the frontend development.
   Available at: https://reactjs.org/docs/getting-started.html

3. **Tailwind CSS Documentation** – Utility-first CSS framework used for responsive styling.
   Available at: https://tailwindcss.com/docs

4. **Node.js Documentation** – Backend runtime environment documentation.
   Available at: https://nodejs.org/en/docs/

5. **Express.js Guide** – Web framework for Node.js used for API development.
   Available at: https://expressjs.com/en/guide/routing.html

6. **MongoDB Documentation** – NoSQL database used for data storage.
   Available at: https://docs.mongodb.com/

7. **JWT Introduction** – JSON Web Tokens for secure authentication.
   Available at: https://jwt.io/introduction

8. **React Query Documentation** – Data fetching and state management library.
   Available at: https://tanstack.com/query/latest

9. **Calisthenics Exercise Guide** – Reference for exercise descriptions and proper form.
   Available at: https://www.calisthenics-gear.com/exercises/

10. **Shadcn/UI Components** – Reusable UI component library.
    Available at: https://ui.shadcn.com/

11. **Framer Motion Documentation** – Animation library for React.
    Available at: https://www.framer.com/motion/

12. **Recharts Documentation** – Composable charting library for React.
    Available at: https://recharts.org/en-US/

13. **MDN Web Docs** – HTML, CSS, and JavaScript reference.
    Available at: https://developer.mozilla.org/

14. **W3Schools** – Web development tutorials and references.
    Available at: https://www.w3schools.com/

15. **Stack Overflow** – Community platform for development problem-solving and technical discussions.
    Available at: https://stackoverflow.com/

END OF DOCUMENTATION