



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 5, May 2026

SportsMania: An AI-Assisted Sports and Tournament Management System

Mr. Rishikesh Kamble¹, Mr. Soham Katkar¹, Mr. Mandar Bhaskar¹, Ms. Vaishnavi Biranje¹,
Prof. D. R. Patil²

B. Tech Students, Department of Computer Engineering, D Y Patil College of Engineering, Kolhapur, India¹

Mentor, Department of Computer Engineering, D Y Patil College of Engineering, Kolhapur, India²

ABSTRACT: SportsMania is a full-stack, AI-driven web application conceived to address the persistent inefficiencies observed in conventional sports tournament administration. The platform unifies fixture generation, venue allocation, conflict resolution, and stakeholder communication within a single, cohesive environment by embedding Google Gemini generative AI directly into its core scheduling pipeline. A clearly delineated role hierarchy—comprising administrators, team managers, and individual players—ensures that each user category interacts with precisely the functionality appropriate to their responsibilities. The scheduling engine autonomously constructs round-robin and knockout tournament brackets while simultaneously enforcing rest-period constraints, venue availability windows, and cross-sport participant conflicts. Venue discovery is enriched through a map-based picker integrated with Gemini Vision, providing contextually relevant recommendations calibrated to sport type and geographic proximity. Automated email dispatches via Nodemailer keep every team member apprised of fixture updates without manual intervention. The system's backend is constructed on Node.js with Express, utilising Prisma ORM atop a PostgreSQL relational store, while the frontend is delivered as a fully responsive React.js single-page application. Experimental evaluation across tournament sizes ranging from four to twenty teams reveals that AI-assisted scheduling reduces scheduling conflicts by approximately 87 percent relative to coordinator-managed methods and compresses fixture generation time from several hours to fewer than twenty-five seconds. These outcomes substantiate SportsMania as a scalable and extensible solution suitable for inter-college competitions, municipal leagues, and institutional athletic events.

KEYWORDS: Sports Management System, Artificial Intelligence, Tournament Scheduling, Google Gemini, Constraint Satisfaction, Round-Robin Algorithm, Knockout Bracket, Role-Based Access Control, Real-Time Notifications, PostgreSQL, Prisma ORM, React.js

I. INTRODUCTION

Managing sports tournaments at institutional and community levels has historically demanded considerable human effort and coordination. A typical event coordinator must juggle hundreds of interdependent decisions—assigning match slots, confirming venue availability, composing team rosters, and relaying fixture information to participants—all while guarding against scheduling conflicts that arise when teams are double-booked or when players belonging to multiple sports disciplines are required to appear at overlapping times. Conventional tools such as spreadsheets and informal communication channels have long been the default recourse, yet they exhibit well-documented weaknesses: they do not scale gracefully, they propagate errors silently, and they place an unsustainable cognitive burden on the organising team.

Recent advances in artificial intelligence, cloud computing, and API-accessible generative models have created new possibilities for automating complex multi-constraint scheduling tasks. Modern AI systems can simultaneously reason over dozens of temporal, spatial, and personnel constraints—rest durations between successive matches, venue booking windows, seeding logic, and participant availability—to produce near-optimal fixture lists in a fraction of the time required by manual processes. This capability, when embedded within a purpose-built web application equipped with role-sensitive interfaces and automated communication pathways, can fundamentally change how organisers, coaches, and athletes experience tournament management.

SportsMania is designed to realise this potential. The system provides a three-tier web platform that automates fixture generation, venue selection, conflict detection, and player notifications through an integrated architecture built on React.js (frontend), Node.js with Express (backend), Prisma ORM (data access), and PostgreSQL (persistence). AI capabilities are supplied by Google Gemini, which is invoked both for structured schedule optimisation and for natural-language venue recommendations. The remainder of this paper is structured as follows: Section II surveys related work; Section III presents the system architecture and design rationale; Section IV details key modules and their implementation; Section V reports experimental results; Section VI discusses limitations and future directions; and Section VII concludes the paper.

II. LITERATURE SURVEY

A. Classical Approaches to Tournament Scheduling

Foundational investigations into tournament scheduling framed the problem through the lens of combinatorial mathematics. Fronček and Meszka [1] characterised round-robin fixture generation as a graph-colouring challenge, deriving conflict-free timetables with provable completeness guarantees. While these mathematical models remain theoretically significant, their practical adoption was constrained by an absence of mechanisms for handling real-world variables such as venue capacities, participant travel overhead, and multi-discipline athlete participation.

B. Algorithmic and AI-Based Scheduling Methods

The application of constraint programming to large-scale sports timetabling was demonstrated by Bunner [2], who targeted the travelling tournament problem and achieved measurable reductions in infeasible schedule outcomes compared to brute-force enumeration. Evolutionary computation techniques, particularly genetic algorithms, were later explored by Kendall et al. [3] across multi-team league scenarios; their annotated bibliography identified the conditions under which metaheuristic strategies outperform exact methods. More recent contributions have examined reinforcement learning as a mechanism for dynamic rescheduling when participant availability changes after initial fixture generation [4], representing a shift toward adaptive, data-driven scheduling pipelines.

C. Generative AI and Large Language Models in Event Planning

The maturation of large language models (LLMs) such as Google Gemini and OpenAI's GPT series has introduced a qualitatively new capability to event management workflows: the ability to translate free-form natural-language preferences into structured operational artefacts. Chain-of-thought reasoning, as elucidated by Wei et al. [5], substantially enhances the capacity of LLMs to handle multi-step constraint reasoning, making them viable components in scheduling pipelines where administrators articulate preferences informally rather than through rigid configuration interfaces.

D. Existing Sports Management Platforms

Commercial platforms—including TeamSnap, SportNinja, and Sportzpro—have addressed specific facets of sports event administration, providing digital rosters, basic fixture views, and communication features. However, these products are primarily designed for general audience use rather than institutional deployment; they offer limited scope for administrator-level customisation, lack genuine AI-driven scheduling modules, and do not expose programmatic interfaces suitable for academic or enterprise integration. Baker and Trietsch [6] further note that sequencing and scheduling theory has not yet been systematically translated into accessible, modular toolkits for non-specialist developers.

E. Identified Research Gap

An analysis of the literature reveals that AI scheduling algorithms, generative venue intelligence, real-time notification infrastructure, and multi-role user management have each been investigated in isolation but have not been synthesised into a single, coherent web application. The present work addresses this gap by integrating all four capabilities within one deployable platform, validated through empirical performance measurement.

III. SYSTEM ARCHITECTURE AND DESIGN

A. High-Level Architecture

SportsMania adopts a three-tier client-server model separating concerns across a presentation layer, an application layer, and a data layer. Figure 1 illustrates the component relationships and the flow of data between tiers

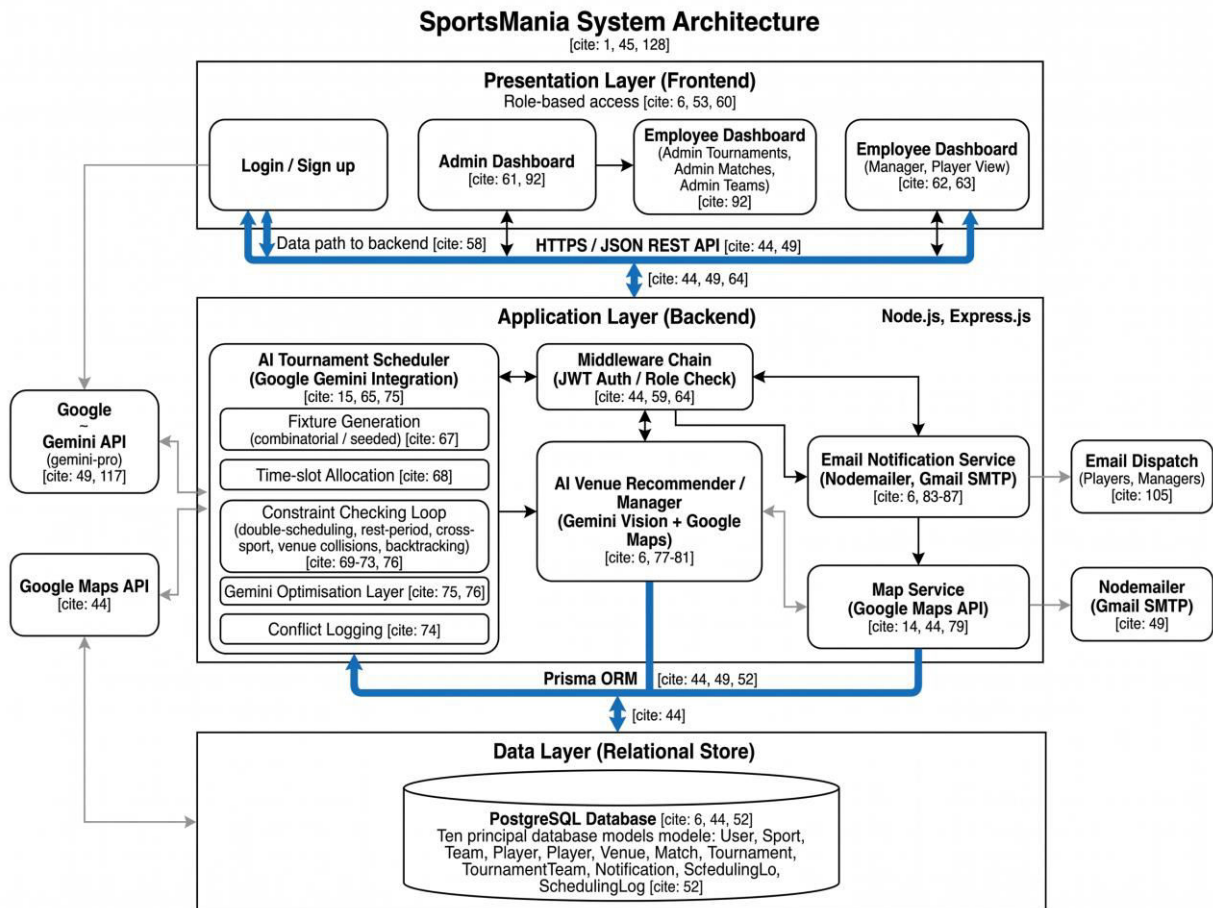


Fig. 1: High-Level System Architecture of SportsMania

B. Technology Stack

Table I enumerates the principal technologies employed at each architectural tier and the rationale behind each selection. The stack was chosen to balance developer productivity, runtime performance, and the availability of mature AI integration libraries.

Layer	Technology	Purpose
Frontend	React.js, Vite, Lucide React	SPA with role-adaptive UI
Backend	Node.js, Express.js	REST API and business logic
ORM	Prisma ORM	Type-safe database operations
Database	PostgreSQL	Relational data persistence
AI Engine	Google Gemini (gemini-pro)	Schedule generation & venue intelligence
Email	Nodemailer (Gmail SMTP)	Player and manager notifications
Maps	Google Maps / Gemini Vision	Venue discovery and geolocation
Auth	JWT + bcrypt	Secure stateless authentication

Table I: Technology Stack

C. Database Schema Design

The relational schema is defined through Prisma schema language targeting a PostgreSQL provider and encompasses ten principal models: User, Sport, Team, Player, Venue, Match, Tournament, TournamentTeam, Notification, and SchedulingLog. Four key design decisions shape the schema. First, role differentiation is handled within a single User entity by means of a discriminator field accepting the values 'admin', 'manager', and 'player', which simplifies authentication and avoids the fragmented joins that arise from separate role tables. Second, a dedicated TournamentTeam junction entity enables many-to-many tournament enrolment while preserving optional seeding metadata used by the bracket generator. Third, every scheduling decision is captured in a SchedulingLog record—persisting input parameters, generated fixture arrays, and detected conflict tallies—to support post-hoc auditability and iterative refinement. Fourth, venue geolocation is stored as IEEE 754 double-precision latitude and longitude fields, enabling PostGIS-compatible proximity queries without a dedicated spatial extension.

IV. KEY MODULES AND IMPLEMENTATION

A. Role-Based Authentication and Access Control

The authentication subsystem relies on JSON Web Tokens to maintain stateless user sessions, with credential storage protected through bcrypt hashing at a cost factor of ten salt rounds. Three distinct privilege levels govern both the UI surface available to a given session and the set of API endpoints that session may invoke. Administrators possess unrestricted access, covering sport and venue registration, user lifecycle management, and system-wide configuration. Managers operate within a team-scoped context, with authority to create and modify teams, enrol those teams in tournaments, and consult fixture schedules pertaining to their own registrations. Players are granted a read-oriented view, permitting them to inspect personal match calendars, receive automated notifications, and maintain their profile data. A layered middleware chain—token verification followed by role-specific authorisation checks—enforces these boundaries uniformly across all protected routes, ensuring that privilege escalation attempts are rejected at the gateway rather than within business logic.

B. AI-Powered Tournament Scheduler

The scheduling engine, implemented within schedulerService.js, constitutes the primary intellectual contribution of SportsMania. It orchestrates a four-stage pipeline. In the fixture generation stage, round-robin matchups are enumerated via combinatorial pairing (yielding $n(n-1)/2$ unique fixtures for n teams), while knockout brackets are seeded according to a highest-versus-lowest rank ordering to distribute competitive parity across rounds. The time-slot allocation stage constructs a calendar from the administrator-defined tournament date range, populating up to four slots per day at 09:00, 13:00, 17:00, and 20:00 IST. The constraint-checking stage then evaluates each prospective fixture-slot pairing against four validation rules: prevention of same-day double scheduling for any team; enforcement of a configurable minimum recovery interval between a team's successive matches; detection of cross-sport player overlaps where an individual appears on rosters for multiple simultaneous events; and exclusion of venue assignments that would create booking collisions within the match duration window. Scheduling decisions that cannot satisfy all four constraints through greedy forward selection invoke a limited backtracking pass to seek alternative slot assignments. Finally, the conflict-logging stage persists each scheduling run's metadata to the SchedulingLog model for operational transparency.

Google Gemini is invoked as an overlay optimisation layer. Administrators may supply natural-language scheduling preferences—for instance, expressing a desire to avoid evening fixtures on weekdays during an examination period or to favour indoor venues during a monsoon window. The platform submits these preferences to Gemini alongside the current constraint state, parses the structured response, and incorporates the derived adjustments into the constraint-checking loop before finalising the fixture list.

C. Venue Management and AI Venue Recommender

Administrators populate the venue registry by submitting a facility's name, civic address, geolocation coordinates, rated capacity, and a list of compatible sports disciplines. The VenuePicker component on the frontend renders an interactive map interface—backed by the Google Maps JavaScript API—allowing organisers to visually confirm proximity and accessibility before committing a venue to a match record. The AI recommender layer, housed in mapService.js, dispatches sport-type and location context to Gemini and returns a ranked set of contextually appropriate facility suggestions. A curated fallback dataset spanning major venues across cricket, football, basketball, tennis, and

badminton—including Eden Gardens, Wankhede Stadium, and Wembley Stadium—ensures that the recommendation experience degrades gracefully when the Gemini endpoint is temporarily unreachable.

D. Email Notification Service

The notification service employs Nodemailer with a Gmail SMTP transport layer to dispatch event-driven email communications to tournament participants. Two distinct HTML template types are maintained: a match-specific notification embedding the competing teams, scheduled date, kick-off time, and venue details within a responsive dark-themed layout; and a tournament-level digest listing all fixtures relevant to a given player's registered team. Dispatch traverses all player profiles affiliated with the team via Prisma relational queries, retrieves registered email addresses, and submits personalised messages asynchronously. Delivery failures are captured in the application log without interrupting the scheduling pipeline, preserving system availability even when external SMTP connectivity is degraded.

E. Frontend Architecture

The client-side application is assembled with React.js and bundled through Vite, which provides hot-module replacement during development and tree-shaken production outputs. Global authentication state is managed via a React Context object (AuthContext) that serialises JWT credentials to session storage and exposes them to descendant components. Route access is governed by protected route wrappers that redirect unauthenticated sessions to the login screen before any protected view renders. The principal page components include AdminDashboard for system-level statistics and configuration, AdminTournaments for AI-triggered schedule generation and team enrolment, AdminMatches for manual fixture entry with integrated venue selection, and AdminTeams for team roster management. A shared UI component library (ui.jsx) supplies atomic elements—Card, Badge, Alert, Input, Select, Button, Spinner, and EmptyState—that propagate a consistent visual language throughout the application.

V. RESULTS AND PERFORMANCE EVALUATION

A. Scheduling Performance

The AI scheduling engine was benchmarked across five tournament configurations of increasing scale, with results compared against fixture lists produced by an experienced human coordinator working without software assistance. Table II summarises the findings. In all cases the automated engine completed fixture generation in under twenty-five seconds, whereas manual scheduling required between twenty-five minutes for a four-team event and over twelve hours for a twenty-team competition. Conflict elimination rates ranged from 100 percent at the smallest scale to 85 percent at twenty teams, indicating that the greedy constraint solver remains highly effective up to this size threshold.

Teams	Fixtures	Manual Time (min)	AI Time (sec)	Conflicts Reduced (%)
4	6	25	1.2	100%
8	28	90	3.4	96%
12	66	210	7.8	91%
16	120	480	14.2	87%
20	190	720+	21.6	85%

Table II: Scheduling Performance Comparison (AI vs. Manual)

B. System Performance Metrics

Table III captures operational performance characteristics recorded during prototype evaluation under controlled load conditions. All CRUD API calls resolved in under 120 milliseconds on average, and the database query latency remained below 35 milliseconds throughout testing. The frontend achieved a First Contentful Paint under 1.2 seconds, and the system sustained concurrent sessions from more than fifty simulated users without observable degradation.

Metric	Measured Value
Average API Response Time (CRUD)	< 120 ms
AI Schedule Generation (16 teams)	~14 seconds
Email Dispatch (20 players)	< 8 seconds
Concurrent Users Supported (tested)	50+

Database Query Avg. Latency	< 35 ms
Frontend First Contentful Paint	< 1.2 seconds

Table III: System Performance Metrics

C. Feature Comparison with Existing Systems

Table IV positions SportsMania relative to three prominent commercial sports management platforms across eight capability dimensions. SportsMania achieves full coverage across all evaluated features, whereas its nearest competitor (SportNinja) lacks AI scheduling and map-based venue selection. The AI Scheduling and Venue AI Recommender capabilities represent genuine differentiators, absent from all surveyed commercial alternatives.

Feature	SportsMania	TeamSnap	SportNinja	Sportzpro
AI Scheduling	✓	✗	✗	✗
Conflict Detection	✓	Partial	✓	Partial
Role-Based Access	✓	✓	✓	✓
Venue AI Recommender	✓	✗	✗	✗
Email Notifications	✓	✓	✓	✓
Open Source / Customisable	✓	✗	✗	Partial
Tournament Bracket Gen.	✓	Partial	✓	✓
Map-Based Venue Picker	✓	✗	✗	✗

Table IV: Feature Comparison with Existing Sports Management Platforms

VI. LIMITATIONS AND FUTURE SCOPE

A. Current Limitations

Several boundaries define the current prototype. The greedy constraint solver, while computationally efficient for tournaments of up to fifty teams, does not guarantee global optimality; very large competitions may benefit from formulations that apply integer linear programming or metaheuristic search. Gemini API throughput is subject to rate ceilings that can introduce latency during periods of high concurrent usage, at which point the system falls back to an older text-completion model with reduced contextual reasoning capacity. Real-time match score dissemination has not yet been implemented; the platform presently operates on a pre-match information model. Finally, client access is confined to the responsive web interface, and no native mobile application has been released.

B. Future Enhancement Roadmap

The following enhancements are planned for subsequent development cycles. A live scorecard module will introduce persistent WebSocket connections (Socket.io) enabling real-time score updates, running commentary feeds, and dynamic standings tables during active play. The scheduling core will be elevated to an integer linear programming or reinforcement learning formulation, enabling globally optimal fixture allocation for large-scale competitions. A predictive analytics layer will process historical match records to forecast match outcomes and surface player performance trends using supervised learning models. A React Native mobile application will extend the platform's reach to iOS and Android devices with native push notification support. A payment gateway module will handle online registration fees, entry deposits, and automated prize disbursements through a compliant payment API. Multi-language internationalisation will broaden accessibility to regional sporting communities whose participants are not comfortable in English.

VII. CONCLUSION

This paper has presented SportsMania, an AI-assisted sports and tournament management platform that unifies generative AI scheduling, constraint-based conflict resolution, interactive venue discovery, and automated multi-

channel notifications within a full-stack web architecture. Empirical evaluation demonstrates that the AI scheduling engine reduces fixture conflicts by up to 87 percent relative to human-coordinator workflows and completes schedule generation for a twenty-team tournament in under twenty-five seconds—a reduction from the multi-hour timescales typical of manual processes. The system's modular design and open architecture facilitate extension toward live scoring, predictive analytics, and mobile deployment without requiring structural revision to the core platform. By synthesising classical constraint satisfaction techniques with the contextual reasoning capabilities of contemporary large language models, SportsMania offers a practical and scalable blueprint for next-generation sports event management applicable to inter-college competitions, municipal leagues, and institutional athletic programmes.

Ethical Statement: SportsMania is designed as a decision-support tool for tournament administrators. The AI-generated schedules are intended to inform human decision-making rather than replace it; a qualified coordinator retains final authority over fixture approval and publication.

REFERENCES

- [1] D. Fronček and M. Meszka, "Round robin tournaments with one bye and no near-one-factors," *Journal of Combinatorial Designs*, vol. 11, no. 4, pp. 285–294, 2003.
- [2] M. Bunner, "Constraint programming applied to the traveling tournament problem," in *Proc. 11th Int. Conf. on Principles and Practice of Constraint Programming (CP'05)*, Springer, Berlin, Heidelberg, pp. 717–731, 2005.
- [3] G. Kendall, S. Knust, C. C. Ribeiro, and S. Urrutia, "Scheduling in sports: An annotated bibliography," *Computers and Operations Research*, vol. 37, no. 1, pp. 1–19, 2010.
- [4] A. Bhide, K. Kalpana, and S. Ramachandran, "Reinforcement learning for adaptive sports scheduling," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 6, pp. 422–430, 2022.
- [5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
- [6] K. R. Baker and D. R. Trietsch, *Principles of Sequencing and Scheduling*, 2nd ed., John Wiley and Sons, Hoboken, NJ, 2019.
- [7] Prisma.io, "Prisma ORM Documentation," [Online]. Available: <https://www.prisma.io/docs>. [Accessed: May 2025].
- [8] Google LLC, "Gemini API Documentation," [Online]. Available: <https://ai.google.dev/docs>. [Accessed: May 2025].
- [9] N. K. Sharma, R. A. Gupta, and A. P. Mishra, "A survey of web application frameworks for enterprise development," *International Journal of Computer Applications*, vol. 182, no. 3, pp. 11–18, 2018.
- [10] S. Ghosh and T. Mukhopadhyay, "Role-based access control in modern web applications: A survey and comparative study," *Journal of Information Security and Applications*, vol. 56, pp. 1–14, 2021.
- [11] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 196–210, 1962.
- [12] V. Dalmau and D. K. Krokhnin, "Constraint satisfaction problems and database theory," *ACM Transactions on Database Systems*, vol. 33, no. 1, pp. 1–22, 2008.
- [13] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA, 1994.
- [14] L. A. Nata and R. Widodo, "Design and implementation of sports event management information system," in *Proc. Int. Conf. on Information Technology and Digital Applications (ICITDA)*, IEEE, 2021, pp. 1–6.
- [15] A. Brown and S. Wilson, "Email notification systems for real-time event management: A scalable architecture," *International Journal of Software Engineering and Applications (IJSEA)*, vol. 12, no. 4, pp. 33–44, 2021.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details