**A PROJECT REPORT**

**ON**

Meetup Platform

**SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE**

**OF**

**BACHELOR OF BUSINESS ADMINISTRATION**

**IN**

**COMPUTER APPLICATION**

**BY**

Vijay Sunil Gholve 5829

**UNDER THE GUIDANCE OF**

Prof. Watpade A.S.

****

**Year 2025-26**

**  **

***Rayat Shikshan Sanstha's***

**S. M. JOSHI COLLEGE, PUNE**

**HADAPSAR, PUNE 411028**

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION**

**2025-26**

**CERTIFICATE**

This is to certify that the project report entitled

Meetup Platform

**-----Submitted by**

Vijay Sunil Gholve 5829

is a bonafide work carried out by them under the supervision of **Prof. Watpade A.S.** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of **Bachelor of Business Administration in (Computer Application)**

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

**Prof. Watpade A.S. Hon. Sangeeta Yadav**

Project Guide Head of Department

Department of Computer Application Department of Computer Application

**Dr. S. T. Salunkhe**

**External Examiner** **Principal**

S. M. Joshi College Hadapsar, Pune

**Place:**

**Date:**

**ACKNOWLEDGEMENT**

We feel great pleasure in expressing our deepest sense of gratitude and sincere thanks to our guide **Prof. Watpade A.S.** for their valuable guidance during the Project work, without which it would have been a difficult task. I have no words to express my sincere thanks for valuable guidance, extreme assistance and cooperation extended to all the Staff Members of our department.

This acknowledgement would be incomplete without expressing our special thanks to **Hon. Sangeeta Yadav,** Head of the Computer Application department for her support during the work.

We would also like to extend our heartfelt gratitude to our Principal, **Dr. S. T. Salunkhe** who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

Last but not the least we would like to thank all the Teaching, Non-teaching staff members of our Department, our Parents and our colleagues those who helped us directly or indirectly for completion of this project successfully.

Vijay Sunil Gholve 5829

**Project Scop**

* Project Scope: ConnectLocal is scoped as a full-stack web application.
* User Management: Comprehensive user registration, profile creation, and
* authentication.
* Group Management: Functionality for creating, joining, leaving, and
* managing groups, including member roles and moderation tools.
* Event Management: Tools for scheduling, promoting, and managing events
* within groups, including RSVP tracking and attendance marking.
* Communication: Integrated messaging features for group members and
* event attendees, alongside discussion boards.
* Discovery: Advanced search, filtering, and basic recommendation capabilities
* for groups and events.
* Basic Analytics: Dashboards for group organizers to view key metrics on
* group activity and event attendance.
* Responsive Design: Optimized for seamless usage across desktop, tablet, and mobile devices

**Limitations**

**Advanced Monetization: While basic event fees might be supported,**

* **complex ticketing tiers or integrated payment processing beyond a single,**
* **primary gateway are out of initial scope.**
* **Live Streaming/Virtual Event Hosting: The platform will primarily focus on**
* **in-person event organization; built-in live streaming or dedicated virtual**
* **event hosting features are not included in the initial release.**
* **Sophisticated AI Recommendations: Advanced, highly personalized AI**
* **driven recommendations based on deep user behavior analysis are**
* **considered for future iterations.**
* **Native Mobile Applications: The project will deliver a fully responsive web**
* **application; dedicated native iOS/Android apps are beyond the current scope.**

**System Analysis**

* **Meetup.com: A long-standing platform specifically designed for local groups and events based on shared interests.**
* **Facebook Groups/Events: A widely used social media feature that allows users to create private or public groups and**
* **organize events.**
* **Eventbrite: Primarily a ticketing and event management platform, often used for larger, public events.**
* **Local Community Forums/Websites: Various smaller, often niche-specific, online forums or websites catering to**
* **particular local interests.**

**Project Perspective, Features**

* Unified Ecosystem A single platform for group discovery, event organization,

communication, and analytics.

* User-Centric Design Prioritizing intuitive navigation and a clean, modern interface for all user roles.
* Robust Group Management Empowering organizers with fine-grained control over their groups and events.
* Enhanced Discovery Intelligent search and filtering to connect users with highly relevant communities.
* Real-time Engagement Integrated chat, discussion forums, and notifications for dynamic interaction.

**Stakeholders**

* Primary Users:
  + Group Organizers: Individuals or entities responsible for creating and managing groups and events.
  + Group Members/Event Attendees: Individuals who join groups and participate in events.
  + Platform Administrators: The team responsible for maintaining the platform, user support, and overall

system health.

* Secondary Stakeholders:
  + Local Businesses/Venues: Who might host events or

offer services to groups.

* + Community Organizations: Non-profits or associations

looking to expand their reach and engagement.

**Functional Requirements**

* User Management: User registration, login/logout, profile creation/editing, password management (reset, change),
* account deactivation.
* Group Management: Create/edit/delete groups, join/leave groups, set group privacy (public/private), manage group
* members (add, remove, assign roles), group discussion forums.
* Event Management: Create/edit/delete events, set event details (date, time, location, description), RSVP functionality
* (attend, decline, maybe), attendee list management, event reminders/notifications.
* Discovery & Search: Search groups by name/interest/location, search events by date/category/location, filter search
* results.
* Communication: In-app messaging between group members, group-wide announcements, event-specific chat.
* Notifications: Real-time notifications for new events, messages, RSVPs, and group updates.
* Reporting & Analytics: Dashboard for organizers to view group member count, event attendance rates, and basic
* engagement metrics.

**Performance Requirements**

* **Response Time:** User interface actions (e.g., loading group pages, submitting RSVPs) should resp Read-Onlyond within 2-3 seconds

under normal load.

* **Scalability:** The system must support concurrent users (e.g., 10,000 active users) and manage a growing number of
* groups and events (e.g., 50,000 groups, 100,000 events) without significant performance degradation.
* **Availability:** The platform should maintain 99.9% uptime, excluding scheduled maintenance.
* **Load Handling:** The system should gracefully handle peak loads during popular event announcements or registration periods.

**Security Requirements**

* Authentication: Secure user authentication using industry-standard protocols (e.g., JWT, OAuth2).
* Authorization: Robust role-based access control (RBAC) to ensure users only access authorized functionalities and data.
* Data Encryption: All sensitive data (e.g., passwords, personal information) must be encrypted both in transit (SSL/TLS)

and at rest (database encryption).

* Input Validation: Comprehensive input validation to prevent common web vulnerabilities (e.g., SQL injection, XSS).
* Privacy Compliance: Adherence to relevant data privacy regulations (e.g., GDPR, CCPA) for user data handling and consent.
* Audit Trails: Logging of critical system actions for security monitoring and incident response.

**System Design**

* Design Constraints
* Budget: Development and operational costs must remain within a defined budget.
* Timeline: Project completion within a specified timeframe (e.g., 6-9 months for initial MVP).
* Technology Stack: Adherence to a modern, scalable, and maintainable technology stack (e.g., Python/Django for backend, React for frontend).
* Scalability: Design must accommodate future growth in user base and data volume.
* Security: Design must inherently incorporate security best practices from the ground up.
* Maintainability: Codebase must be clean, modular, and well-documented for future enhancements and bug fixes.

**User Interfaces**

* The user interfaces for ConnectLocal will be designed with a strong emphasis on intuitiveness, responsiveness, and consistency.
* Intuitive Navigation: Clear and logical navigation paths will guide users effortlessly through the platform, minimizing learning curves.
* Responsive Design: The UI will dynamically adapt to various screen sizes (desktop, tablet, mobile) using a mobile-first approach, ensuring optimal usability
* across all devices.
* Consistent Experience: A unified design language, including consistent typography, color palettes, and component styling, will be applied across the entire
* platform for a cohesive user experience.
* Accessibility: Adherence to WCAG (Web Content Accessibility Guidelines) will ensure the platform is usable by individuals with disabilities, incorporating
* features like keyboard navigation, proper color contrast, and screen reader compatibility.
* Visual Appeal: A clean, modern, and engaging aesthetic will be employed to enhance user satisfaction and encourage prolonged engagement.

DFD LEVEL 0

**DFD LEVEL 1**

**ER -DIAGRAM**

**Activity Diagram**

**Collaboration diagram**

**sequence diagram**

**Class diagram**

**UseCase Diagram**

**Software/Hardware Specifications**

**Backend:**

* Language/Framework: Python 3.x with Django (latest stable version).
* API Framework: Django REST Framework.
* Database: PostgreSQL (for scalability and robust features).
* Authentication: Django's built-in authentication system with JWT for
* API token management.
* Image/File Handling: Pillow library for image processing.
* Real-time Communication: Django Channels (for WebSockets) or a
* dedicated messaging queue (e.g., Redis Pub/Sub).
* Deployment Environment: Cloud platform (e.g., AWS EC2/ECS, Google
* Cloud Run, Azure App Service) with containerization (Docker)

**Frontend:**

* **Language/Library: JavaScript with React (latest stable version).**
* **Build Tool: Vite.**
* **Routing: React Router DOM.**
* **State Management: Redux Toolkit.**
* **HTTP Client: Axios.**
* **UI Framework/Libraries: Tailwind CSS for utility-first styling,**
* **potentially Material-UI or Ant Design for robust components, Framer**
* **Motion for animations, Lucide React for icons**