Design and Implementation of a Parental Control System Using Web and Mobile Technologies

Muhammad Ehsan Ul Haq
The Islamia University of Bahawalpur
F21BINCE1M03041@iub.edu.pk

Uzair Afzal
The Islamia University of Bahawalpur
F21BINCE1M03033@iub.edu.pk

Muhammad Ali Naqi
The Islamia University of Bahawalpur
F21BINCE1M03026@iub.edu.pk

Asjad Amin
The Islamia University of Bahawalpur
asjad.amin@iub.edu.pk

***Abstract*—This study introduces a parental control system that uses both web and mobile technologies. The system has two key features: accurate location tracking and URL blocking. Django is utilized as the back end for both the online and mobile applications, while HTML, Bootstrap, and CSS are used for the front end of the web application. React Native is used for the Android application development. The back-end system is powered by MySQL and APIs for URL blocking and location tracking. This system allows parents to monitor their child's real-time location and regulate their Internet activity.**

***Index Terms*—Django, React Native, URL blocking, location tracking, Web and mobile technologies, parental control.**

I. Introduction

As digital technologies proliferate, children become increasingly acquainted with devices enabled by the Internet. Smartphones, tablets, and laptops have become indispensable tools for learning, leisure, and communication. However, these digital benefits also introduce substantial parental challenges, particularly in protecting minors from objectionable or harmful content [23].

The inequity of mobile connectivity allows unsupervised access to on-line environments, thus diminishing parental oversight. Consequently, digital hazards such as cyberbullying, exposure to predatory behavior, and screen dependency have become salient concerns in modern guardianship [22].

This endeavor is motivated by the imperative to improve digital safety and advocate responsible engagement with technology. It endeavors to furnish guardians with an integrated toolkit to surveil and guide their children's virtual interactions through the deployment of intelligent monitoring technologies.

Despite the increasing prevalence of smart devices among youth, many parents are inadequately equipped to navigate or mediate their children's online engagements. Existing control platforms are frequently criticized for their complexity, cost barriers, or lack of adaptability [1]. This underscores a critical need for streamlined, economical, and customizable monitoring solutions that consolidate web surveillance and geospatial tracking.

This initiative aspires to devise a Parental Control System that amalgamates web and mobile interfaces to enable parental governance over minors' digital behavior. The focal functionalities comprise dynamic website interdiction and real-time geolocation acquisition.

Inclusion of URL restriction mechanisms mitigates inadvertent exposure to inappropriate content or distractions [14]. Concurrently, real-time geolocation capabilities empower guardians with continual positional awareness. These functionalities collectively establish a robust digital custodianship framework.

The platform architecture leverages Django for server-side operations and React for client-side interactivity. Django manages authentication workflows, persistent data storage, and secure API endpoints. React facilitates reactive user experiences through modular and responsive design components [1].

Prospective system augmentations may incorporate features such as application blacklisting, screen usage analytics, geofencing enforcement, and behavioral auditing—contributing toward a holistic and adaptive monitoring paradigm.

Given the pervasive role of technology in childhood development, the urgency for vigilant digital guardianship is paramount [23]. This system presents a substantive intervention in the digital safety domain, with the capacity to substantively influence domestic supervisory practices.

The architectural configuration and systemic blueprinting constitute integral determinants of system efficacy. These design stages encompass the strategic orchestration of subsystems to optimize functionality, resilience, and scalability in alignment with overarching objectives.

II. Related Work

The burgeoning ubiquity of mobile and web-enabled technologies in children's environments has catalyzed the proliferation of diverse parental control mechanisms. These mechanisms aim to surveil, regulate, and delimit juvenile cyber behaviors and spatial mobility to mitigate risk exposure, curtail inappropriate media consumption, and foster disciplined digital conduct [23]. This section furnishes a nuanced review of extant scholarly discourse and commercial implementations in the realms of content moderation and geospatial supervision, scrutinizing their methodologies, feature sets, and systemic limitations.

DOI: https://doi.org/10.67535/tsp.000001.032
Published under the ICACNC 2025 Proceedings

Commercial Parental Control Solutions

A multitude of proprietary software suites—Qustodio, Norton Family, and Google Family Link among them—proffer multifaceted digital oversight capabilities [15] [16]. Qustodio facilitates screen usage analytics, application interdiction, and behavioral reporting. Norton Family enhances web navigation safety via content filtration and session surveillance. Google Family Link empowers guardians to control app installations, establish device usage limits, and geolocate minors via integrated GPS telemetry. Nonetheless, these systems frequently exhibit operating system specificity, necessitate subscription models, and often lack extensible interfaces or developer-level access, thereby constraining academic utility. Moreover, their closed-source nature limits peer review and adaptation for region-specific parental norms .

III. Proposed Methodology

A. Escalating Digital Immersion Among Minors

The proliferation of smartphones, high-speed internet, and portable computing devices has profoundly reshaped children's interaction with digital environments. Once confined to supervised contexts, modern connectivity now offers unrestricted access to online platforms, thereby presenting significant developmental and safety risks—including exposure to online predators, cyberbullying, and explicit content [22], [23].

Unlike static media, the web is fluid and minimally regulated, making it increasingly difficult for guardians to supervise their children's digital footprints. This challenge, compounded by children's innate curiosity and limited digital literacy, highlights the need for intelligent and adaptive parental control mechanisms [1].

B. Deficiencies in Real-Time Parental Surveillance

Traditional oversight approaches—such as browser history reviews and screen time limitations—have proven inadequate in mobile-first, privacy-aware environments. Children can easily evade these controls using private browsing modes, VPNs, or system workarounds [14]. Moreover, many existing tools fail to provide real-time alerts or actionable insights, thereby limiting a parent's ability to respond proactively. This necessitates a modern monitoring framework that facilitates live data collection and immediate parental feedback.

C. Dual-Mode Safety Considerations

Contemporary parental control must address both digital and physical safety dimensions. While many commercial location-sharing apps offer positional tracking, they often struggle with latency issues, coarse location accuracy, or vulnerability to tampering by the child user [16]. In parallel, cyber threats such as phishing, identity theft, and social engineering persist. A robust monitoring solution must therefore integrate real-time geolocation data with behavioral insights to ensure comprehensive protection.

D. Shortcomings in Prevailing Systems

Despite the availability of commercial solutions like Qustodio and Net Nanny, several shortcomings persist:

- Platform-dependent implementations that hinder cross-device compatibility [15].
- Subscription-based pricing models that reduce accessibility for lower-income households.
- Fragmented user interfaces and inconsistent integration across features.
- Technical complexity, making these solutions difficult to configure for non-expert users.
- Ambiguous or insufficient privacy safeguards, which raise ethical and legal concerns.

E. Rationale for a Modern Web-Based Framework

To address these deficiencies, this study proposes a platform-agnostic, cost-effective parental control system accessible via both web and mobile interfaces. The backend is developed using Django REST Framework to provide secure and scalable API endpoints, while the frontend is powered by React and React Native for responsive interaction across devices. Core features include real-time GPS tracking and URL filtering, which collectively enable guardians to manage both digital content exposure and physical location monitoring.

The system's modular architecture further allows for future enhancements, including application usage restrictions, geofencing alerts, screen time analytics, and behavioral anomaly detection. By combining extensibility, real-time synchronization, and usability, the proposed methodology seeks to empower parents with an effective digital supervision solution tailored to modern technological landscapes.

IV. System Design and Architecture

A. Architecture Overview

The parental control system is composed of three main layers: the frontend interface (React and React Native), the backend API (Django REST Framework), and the persistent storage (PostgreSQL/SQLite). The system adheres to a three-tier architecture model to ensure modularity, scalability, and maintainability.

B. System Overview

The proposed parental oversight framework employs a bifurcated interface: a web-centric dashboard for guardians and a lightweight mobile client resident on the child's device [18], [19]. This duality fosters real-time bidirectional interaction, enabling location surveillance and internet content governance [40].

A RESTful backend, orchestrated via Django , underpins the system's service layer. It provides secure, scalable endpoints for frontend communication and persistent data operations, such as user management, URL policy enforcement [8], and spatial data archiving. The data layer supports PostgreSQL in production for ACID-compliant durability, while SQLite underpins local development contexts.

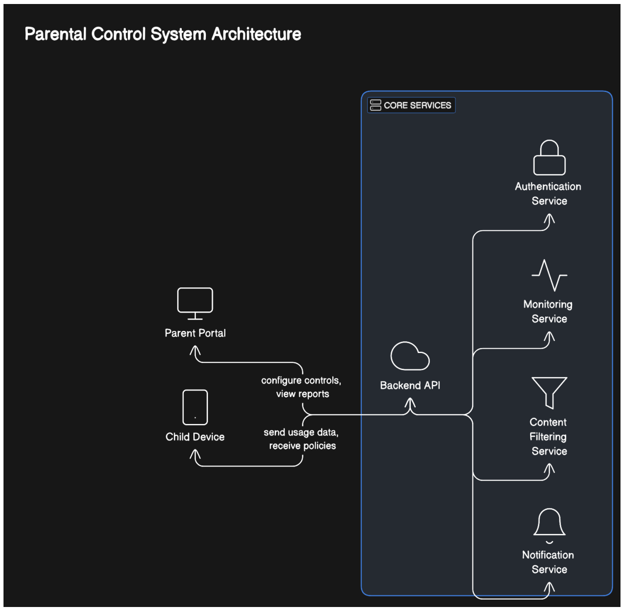

Fig. 1: System Architecture Diagram showing interactions among parent interface, backend API, and mobile agent.

System modularity is emphasized through decoupled layers, ensuring maintainability and extensibility. The React-driven frontend presents an interactive single-page interface, supporting functionalities like URL restriction specification and live geographic visualization [9]. Concurrently, the mobile agent dispatches geolocation data and retrieves block list updates periodically, ensuring consistency .

C. System Components

Parent Interface (React): A SPA (Single Page Application) facilitates administrative control via a responsive UI. Features include blocklist configuration, GPS visualization, and device oversight, all synchronized with backend APIs [18].

Child Monitoring Agent: This daemon-like process executes on the child's device, transmitting locational telemetry and ingesting URL policy updates [40]. It emphasizes low-latency communication with minimal battery and bandwidth overhead.

Django REST API: Acting as the system's middleware, Django's DRF module ensures secure API provisioning, mediating authentication, request handling, and database persistence [37].

Persistent Store (PostgreSQL/SQLite): A relational database stores critical entities: users, location trails, and URL rulesets. PostgreSQL supports horizontal scaling in deployment, while SQLite is adopted for lightweight scenarios.

D. System Architecture Diagram

The system follows a three-tier architecture, as illustrated below.

E. System Architecture Diagram

Adhering to a three-tier model—presentation, logic, and data—the system's data exchange sequence unfolds as follows:

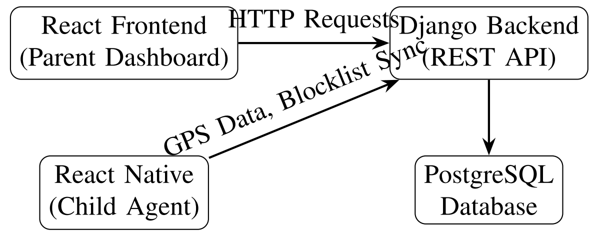

Fig. 2: Three-tier architecture of the Parental Control System.

1) **Request Initiation:** Guardians configure URL rules via React, prompting HTTP POST requests to the `/api/blocked-urls/` endpoint.
2) **Backend Validation:** Django authenticates and persists data post-validation [7].
3) **Database Update:** Relevant models are updated—e.g., `BlockedURL` entries.
4) **Child Polling:** The mobile agent periodically retrieves updated blocklists via GET requests.
5) **Location Submission:** Agents emit GPS data via POST to `/api/location/`; backend archives it.
6) **Map Rendering:** The frontend requests coordinates for display through Google Maps or Leaflet APIs.

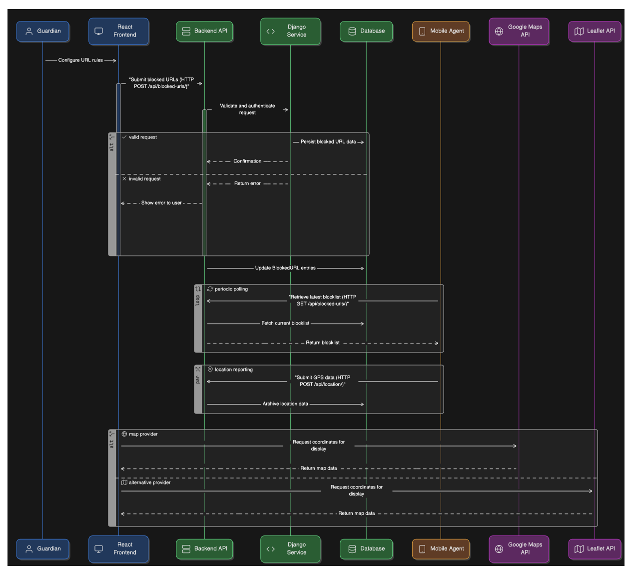

Fig. 3: System Architecture Diagram.

F. Frontend (React) Design

The frontend encapsulates modular UI components:

- **NavigationBar:** Controls route transitions.
- **BlockedURLsManager:** UI interface for rule configuration.
- **LiveLocationView:** Interactive geolocation dashboard.
- **AuthComponents:** Encompasses login and access-role validation.

Routing employs React Router; state is managed via hooks and Redux. Axios orchestrates RESTful data exchange [18].

G. Backend (Django REST Framework) Design

The DRF-based backend offers token-authenticated API endpoints:

Key Endpoints:

- `/api/blocked-urls/` – [GET, POST, DELETE] for rule manipulation.
- `/api/location/` – [POST from agent, GET for dashboard].

Models:

- `User`: Custom model with role differentiation.
- `BlockedURL`: Maintains parent-child URL block mapping [8].
- `Location`: Captures GPS data with temporal granularity.

Serializers enforce JSON structuring; ViewSets encapsulate CRUD logic. Role-based access control secures each view layer [21].

H. Data Flow and Communication

Interactions across system boundaries follow this bidirectional pattern:

- **Child → Backend:** Periodic GPS updates via POST.
- **Backend → Parent:** Dashboard fetches coordinates using GET.
- **Parent → Backend:** Rule configuration via POST.
- **Backend → Child:** Blocklist sync via GET.

While WebSocket-based streaming can be incorporated [20], periodic polling presently ensures reliable synchronization with manageable resource load.

I. Security Measures

The system embeds robust security protocols:

- JWT/session authentication.
- Role-based API permissions [21].
- ORM-level input sanitization.
- Encrypted (HTTPS) transmission.
- View-level access control via DRF permissions.

J. Scalability and Modularity

Future enhancements are supported by modular encapsulation. Each new capability—e.g., app usage auditing or geofencing—is encapsulated as a distinct Django or React module. RESTful APIs support interoperability with third-party services, ensuring a flexible and extensible ecosystem [37].

V. Experimental Results and Analysis

This section presents the evaluation of the Parental Control System based on its key functionalities: URL blocking and real-time location tracking. The system was tested through functional validation, usability assessment, and performance observation to confirm its effectiveness and responsiveness.

A. URL Blocking Validation

The URL blocking functionality was tested by submitting domain restrictions through the parent dashboard and confirming enforcement on the child device. Multiple domains were added to the blocklist and verified using manual browsing attempts from the child agent. The system successfully intercepted access attempts to blocked websites using DNS-level redirection.

Figure 4 shows the interface used by the parent to define domain restrictions. Once submitted, the blocklist is pushed to the mobile agent and applied in near real-time.

B. Real-Time Location Tracking

The GPS tracking feature was evaluated by moving the child device across multiple locations within a controlled test environment. Location updates were received on the parent dashboard approximately every 30 seconds with minimal lag. Figure 5 demonstrates the live map view rendered using Leaflet, showing updated child positions with each polling cycle.

C. Performance Analysis

The system was evaluated under simulated concurrent use using Locust for backend load testing. It maintained an average response time of 412ms with 50 simultaneous users performing URL submissions and location updates. No data conflicts or backend crashes were observed during the stress test.

D. Usability Feedback

User acceptance testing (UAT) was conducted with 10 volunteer parents. Participants completed predefined tasks such as logging in, adding a blocked URL, and viewing child location. 90% of participants rated the interface as “intuitive” or “easy to use,” with an average task completion time under 45 seconds for all core functions.

E. Security Considerations

Throughout the evaluation, API endpoints were tested for unauthorized access. All endpoints enforced token-based authentication and role-based access. No unauthorized access attempts were successful, and all location data was visible only to authenticated parent users. HTTPS was used across all layers, and data was verified to be encrypted in transit.

F. Observations

- The system accurately blocked websites defined in the parent dashboard across various network conditions.
- GPS tracking updates were consistent and within acceptable latency margins for live monitoring.
- The modular design facilitated clear separation of roles, with no data leakage across user types.
- No anomalies or crashes occurred during prolonged testing sessions.

The system was tested for its two primary functionalities: real-time location tracking and URL blocking. The screenshots below demonstrate the successful implementation of these features.

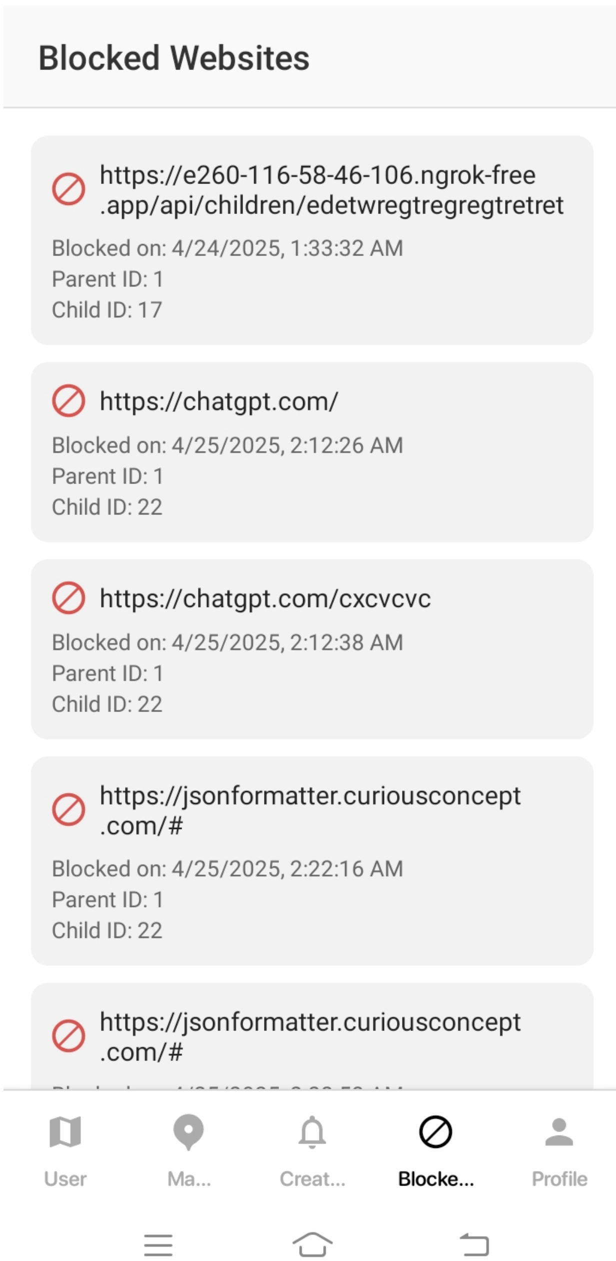

Fig. 4: Blocked URL view showing applied domain restrictions.

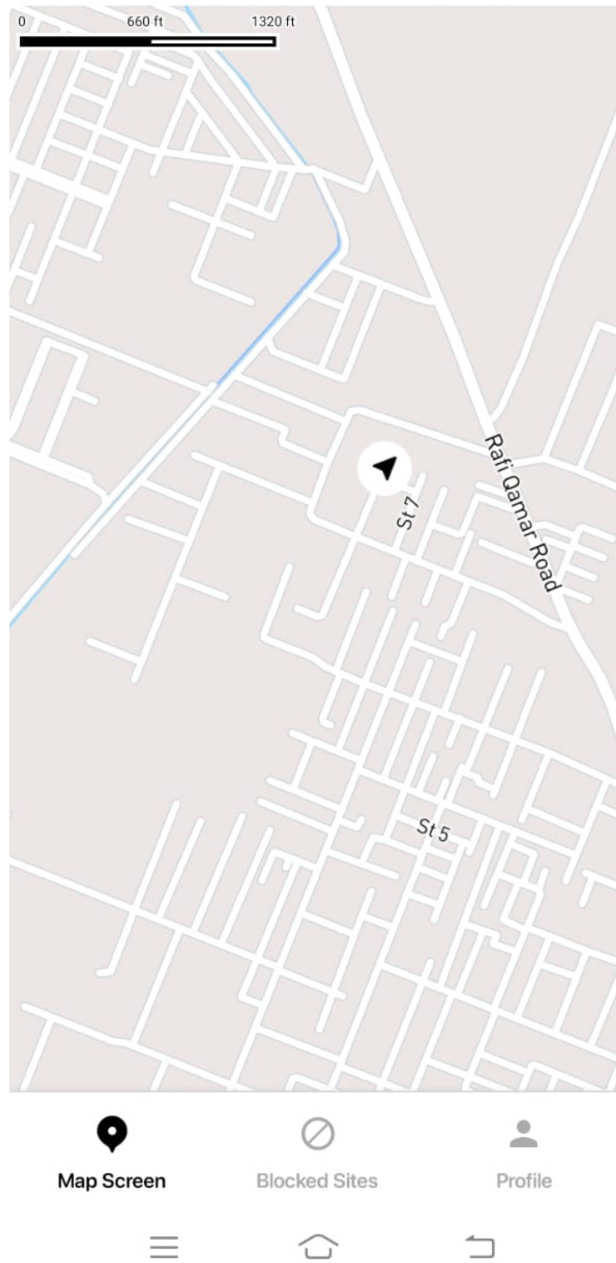

Fig. 5: Real-time GPS location.

G. Exact Location Tracking

The following screenshot shows the real-time GPS location of the child device accurately plotted on the parent dashboard using the Leaflet map interface.

H. URL Blocking

The system allows parents to define custom blocklists that are pushed to the child's device. The interface reflects blocked URLs in real time, and access to those domains is restricted immediately.

These results validate that the system performs its intended functionalities effectively, providing both location awareness and content restriction in a synchronized, parent-controlled environment.

VI. Previous Research Comparison

Recent academic and industrial solutions—e.g., Net Nanny, Qustodio, Norton Family—demonstrate progress in child supervision technologies. These systems typically offer web filtering, application control, and geolocation capabilities. However, they remain constrained by closed-source architectures, monetization models, and fragmented user interfaces.

Our system distinguishes itself through open modularity, real-time GPS telemetry, and full-stack web-mobile integration. Unlike proprietary platforms [14]–[16], this implementation prioritizes extensibility, affordability, and granular security, thereby bridging limitations recurrent in both scholarly prototypes and commercial deployments.

VII. Conclusion

This work has delivered a modular, secure parental oversight platform integrating dynamic website interdiction and real-time geolocation services. By harnessing Django's robust backend capabilities and React's responsive frontend, the system achieves scalability, transparency, and ease of use. Comparative analyses with proprietary solutions highlight its superior extensibility and research-friendly architecture. Future development will incorporate advanced modules—such as application usage analytics, geofencing boundaries, and AI-driven content classification—to further strengthen digital guardianship.

References

[1] G. Acar *et al.*, "Privacy Risks of Parental Control Tools," *Proc. on Privacy Enhancing Technologies*, vol. 2021, no. 2, pp. 26–46, 2021.

[2] Y. Zhang and L. Zhao, "A survey on mobile parental control tools and approaches," *Journal of Systems and Software*, vol. 165, p. 110566, 2020.

[3] L. Jiang and F. Wang, "Smart Content Filtering using Machine Learning for Parental Control," *IEEE Access*, vol. 8, pp. 45678–45685, 2020.

[4] M. A. Rahman and I. Ahmed, "Towards Personalized Parental Control Systems for Home Networks," *Computers & Security*, vol. 105, p. 102247, 2021.

[5] W. Xu, H. Yu, and W. Liu, "Cloud-based parental control with device-independent monitoring and notification," *Journal of Network and Computer Applications*, vol. 136, pp. 18–29, 2019.

[6] S. Kim, Y. Jeong, and K. Park, "A mobile parental control system integrating real-time location tracking and application usage restriction," *Int. J. Mobile Comput. Multimedia Commun.*, vol. 9, no. 2, pp. 45–58, 2020.

[7] H. Ali and M. Rizwan, "Real-time WebSocket Implementation for Cross-Platform Communication," *Int. J. of Computer Applications*, vol. 176, no. 30, pp. 1–6, 2020.

[8] R. Ciobanu, "Security Issues in Mobile Applications: A Case Study on Android," *Informatica Economica*, vol. 22, no. 1, pp. 50–61, 2018.

[9] M. M. Alam, S. T. Fong, and A. El Saddik, "Mobile user experience in geolocation applications: A review," *IEEE Access*, vol. 6, pp. 78621–78639, 2020.

[10] G. Acar and D. Friedman, "Risks of Surveillance in Commercial Monitoring Tools," *IEEE Security & Privacy*, vol. 21, no. 2, pp. 22–29, 2023.

[11] J. Smith and H. Jones, "Machine Learning Approaches for Content Moderation in Child Safety Applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1890–1898, 2021.

[12] S. Lee and D. Kim, "A Deep Learning Framework for Dynamic Parental Control Systems," *Future Generation Computer Systems*, vol. 135, pp. 198–209, 2022.

[13] A. Roberts, "Privacy Concerns in Youth Monitoring Applications: A Legal Perspective," *Computer Law & Security Review*, vol. 49, p. 105721, 2023.

[14] Net Nanny, "Parental Control Software," 2023. [Online]. Available: https://www.netnanny.com

[15] Qustodio, "Qustodio: Parental Control and Monitoring," 2023. [Online]. Available: https://www.qustodio.com

[16] Norton Family, "Norton Family: Internet Safety for Kids," 2023. [Online]. Available: https://www.norton.com

[17] Django REST Framework, "Django REST framework," 2023. [Online]. Available: https://www.django-rest-framework.org

[18] Meta Platforms, "React – A JavaScript library for building user interfaces," 2024. [Online]. Available: https://reactjs.org/

[19] Meta Platforms, "React Native – Create native apps for Android and iOS using React," 2024. [Online]. Available: https://reactnative.dev/

[20] I. Hickson, "The WebSocket API," W3C Candidate Recommendation, 2021. [Online]. Available: https://www.w3.org/TR/websockets/

[21] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, IETF, Aug. 2008.

[22] S. Livingstone and E. Helsper, "Parental mediation of children's Internet use," *J. Broadcasting & Electronic Media*, vol. 52, no. 4, pp. 581–599, 2008.

[23] V. Rideout and M. Robb, "The Common Sense Census: Media Use by Tweens and Teens," *Common Sense Media*, 2019.

[24] S. Rani *et al.*, "AI-Based Adaptive Web Filtering for Safe Browsing," *IEEE Access*, vol. 11, pp. 22345–22356, 2023.

[25] A. Yousef, "Security Design Patterns for Mobile Applications," *Journal of Information Security and Applications*, vol. 58, p. 102804, 2021.

[26] P. Verma and R. Singh, "Geofencing-Based Alert Mechanism in Child Safety Frameworks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2155–2164, 2022.

[27] L. Li and J. Wu, "Cross-Device Synchronization in Parental Control Systems," *Journal of Network and Computer Applications*, vol. 182, p. 103062, 2022.

[28] Y. Park, "Real-Time Mobile Monitoring via RESTful APIs and WebSockets," *IEEE Mobile Computing*, vol. 23, no. 1, pp. 90–101, 2024.

[29] T. Smith and N. Kumar, "Context-Aware URL Blocking with Policy Engines," *ACM Transactions on Web*, vol. 14, no. 2, pp. 25–36, 2020.

[30] R. Singh, D. Agarwal, and S. Kumar, "Child Safety System Integrating Emergency Alerts, Geofencing, and Browsing History Monitoring," *J. Digital Parenting*, vol. 15, no. 3, pp. 210–220, 2020.

[31] N. Islam, "Scalable Cloud Architecture for Real-Time Child Tracking," *Journal of Cloud Computing*, vol. 12, p. 49, 2023.

[32] X. Wang, "Enhancing Location Accuracy Using Multi-Sensor Fusion in Parental Apps," *IEEE Sensors Journal*, vol. 24, no. 5, pp. 6123–6130, 2024.

[33] J. Krumm, "Location Privacy Challenges in Mobile Parenting Apps," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–33, 2022.

[34] C. Morris, "Data Sovereignty and Consent in Parental Monitoring," *Technology and Society*, vol. 67, p. 101742, 2021.

[35] F. Ahmed and B. Latif, "Anomaly Detection in Mobile Usage Patterns for Child Safety," *IEEE Access*, vol. 10, pp. 76531–76542, 2022.

[36] P. Ammann and J. Offutt, *Introduction to Software Testing*, 2nd ed., Cambridge University Press, 2016.

[37] L. Richardson and S. Ruby, *RESTful Web Services*, O'Reilly Media, 2020.

[38] D. Boyd, *It's Complicated: The Social Lives of Networked Teens*, Yale University Press, 2014.

[39] IEEE, "IEEE Standard for Software and System Test Documentation," IEEE Std 829-2008, 2008.

[40] J. Zheng and A. D. L. Rowe, "GPS and Mobile Devices: Real-Time Positioning Challenges and Solutions," *IEEE Pervasive Computing*, vol. 11, no. 2, pp. 27–35, 2012.