

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum - 590014, Karnataka

Full Stack Web Development Report On "ECO-SEVAKS"

Environmental Volunteering Platform

Submitted by:

NAME USN

[Student Name 1] [USN 1]

[Student Name 2] [USN 2]

[Student Name 3] [USN 3]

[Student Name 4] [USN 4]

Under the Guidance of

[Faculty Name]

[Designation]

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous College affiliated to VTU)

BENGALURU - 560 019

Sep 2024 - Jan 2025

CERTIFICATE

B.M.S. College of Engineering

Department of Computer Science and Engineering

This is to certify that the project work entitled "**ECO-SEVAKS**" is a bonafide work carried out by **[Student Name 1] ([USN 1]), [Student Name 2] ([USN 2]), [Student Name 3] ([USN 3]), [Student Name 4] ([USN 4])**, students of B.M.S. College of Engineering, Bengaluru, during the year 2024-2025, in respect of the course **Full Stack Web Development (23CS3AEFWD)**.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements prescribed for the said course.

Signature of the GuideName: _____ Name: _____
Date: _____ Date: _____**Signature of HOD****External Viva**

DECLARATION

B.M.S. College of Engineering

Department of Computer Science and Engineering

We, **[Student Name 1] ([USN 1]), [Student Name 2] ([USN 2]), [Student Name 3] ([USN 3]), [Student Name 4] ([USN 4])**, students of B.M.S. College of Engineering, Bengaluru, VIII Semester, Department of Computer Science and Engineering, hereby declare that the project work entitled "**ECO-SEVAKS**" has been carried out by us in B.M.S. College of Engineering, Bengaluru, and submitted in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum.

We further declare that the project work is original and has not been submitted to any other university/institution for the award of any other Degree/Diploma/Certificate.

Signatures:

[Student Name 1] [Student Name 2]
[Student Name 3] [Student Name 4]

Place: Bengaluru**Date:** _____

TABLE OF CONTENTS

SL. NO.	TITLE	PAGE NO.
1	Introduction	5

1.1 Overview	5
1.2 Motivation	5
2 Software Requirement Specification	6
2.1 Hardware Requirements	6
2.2 Software Requirements	6
3 ER Diagram of the Project	7
4 Schema of the Project	8
5 User Interface Design	9
6 Back End Design	10
7 Conclusion and Future Work	11
8 References	12

1. INTRODUCTION

1.1 OVERVIEW

ECO-SEVAKS is a comprehensive full-stack web application designed to connect environmental volunteers with eco-friendly events and initiatives across India. The name "Sevaks" (सेवक) translates to "servants" in Hindi, embodying our mission to serve Mother Earth through coordinated environmental action.

The platform serves as a digital ecosystem where environmentally conscious citizens can discover, organize, and participate in environmental conservation activities. Built using cutting-edge web technologies including **React.js** for the frontend, **Node.js with Express.js** for the backend, and **MongoDB** as the database, ECO-SEVAKS provides a seamless, responsive, and intuitive user experience across all devices.

Key functionalities of ECO-SEVAKS include:

- **Event Discovery:** Users can browse approved environmental events such as tree plantation drives, beach cleanups, recycling workshops, and sustainability awareness campaigns.
- **Event Creation:** Registered users can create and submit their own environmental events for admin approval.
- **RSVP Management:** Volunteers can register for events with one-click, track their registrations, and cancel if needed.
- **Admin Dashboard:** Administrators can review pending events, approve or reject submissions, and manage the platform.
- **User Dashboard:** Personalized dashboards showing joined events, created events, and participation statistics.
- **Volunteer Management:** Event organizers can view and manage registered volunteers for their events.

The application implements **role-based access control** with two primary roles: Users (who can both volunteer and organize events) and Administrators (who oversee event approvals and platform management). Session-based authentication ensures secure access to protected features.

1.2 MOTIVATION

The motivation behind developing ECO-SEVAKS stems from several critical environmental and social challenges:

1. Environmental Crisis in India:

India faces severe environmental challenges including air pollution, plastic waste accumulation on beaches and public spaces, deforestation, and water contamination. According to environmental studies, coordinated volunteer action can significantly impact local ecosystems and community health.

2. Fragmented Volunteering Ecosystem:

Currently, environmental volunteering opportunities are scattered across social media platforms, WhatsApp groups, and individual organization websites. This fragmentation makes it difficult for willing volunteers to discover events and for organizers to reach their target audience effectively.

3. Need for Digital Coordination:

Traditional methods of organizing environmental initiatives require substantial coordination effort through phone calls, manual registrations, and physical meetings. A centralized digital platform eliminates these barriers and enables efficient event management.

4. Youth Engagement:

With increasing environmental awareness among younger generations, there is growing demand for accessible, tech-friendly platforms that enable youth participation in conservation activities. Social media promotions often fail to provide proper registration and tracking mechanisms.

5. Transparency and Accountability:

Many environmental initiatives lack proper attendee management, making it difficult to measure impact or ensure volunteer safety. ECO-SEVAKS addresses this through event approval workflows, capacity management, and detailed volunteer tracking.

Benefits Provided:

- **For Volunteers:** Easy event discovery, seamless registration, personal impact tracking
 - **For Organizers:** Broader reach, simplified volunteer management, reduced coordination overhead
 - **For Environment:** More coordinated conservation efforts leading to measurable positive impact
-

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

Component	Specification
Processor	Intel Core i5 (8th Gen) or higher / AMD Ryzen 5 equivalent
RAM	Minimum 8 GB DDR4 (16 GB recommended)
Hard Disk	256 GB SSD with minimum 10 GB free space
Display	1920 x 1080 resolution or higher
Network	Stable internet connection (minimum 10 Mbps)

For End Users:

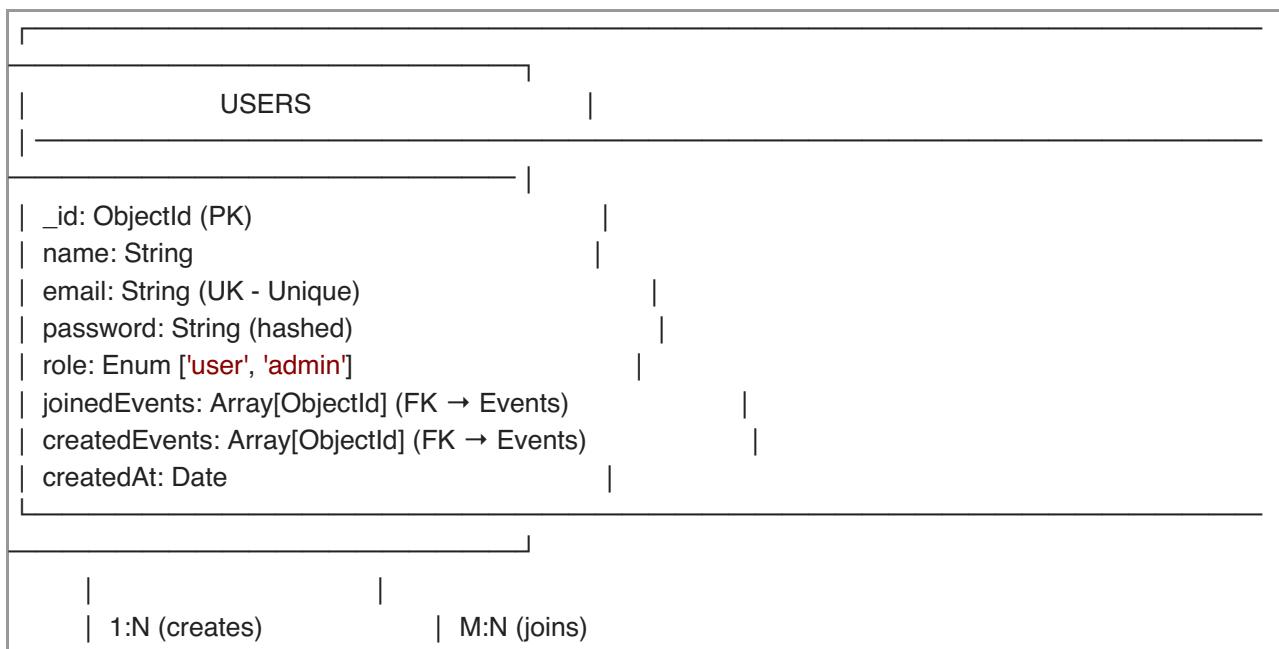
- Any modern computer, tablet, or smartphone
- Minimum 2 GB RAM
- Updated web browser (Chrome, Firefox, Safari, Edge)
- Internet connectivity (3G/4G/5G/WiFi)

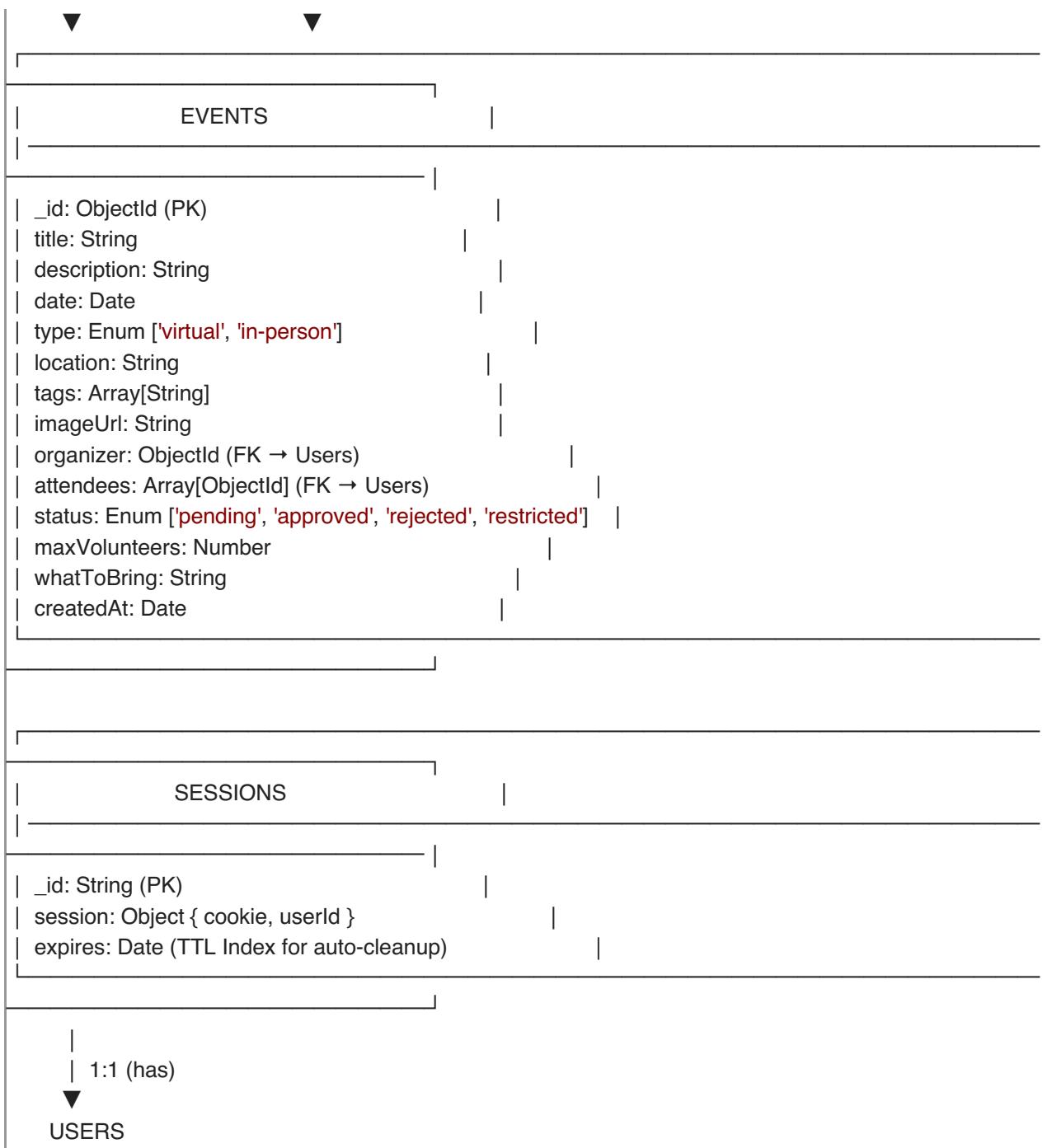
2.2 Software Requirements

Category	Technology	Version
Operating System	Windows 10/11, macOS, Ubuntu	Linux Latest
Frontend Framework	React.js	18.2.0
Build Tool	Vite	7.2.1
CSS Framework	Tailwind CSS	3.4.3
HTTP Client	Axios	1.13.2
Icons	Lucide React	0.378.0
Notifications	React Hot Toast	2.6.0
Backend Runtime	Node.js	18.x LTS
Backend Framework	Express.js	4.18.2
Database	MongoDB (Native Driver)	6.10.0
Password Hashing	bcryptjs	2.4.3
Session Management	express-session	1.18.2
Session Store	connect-mongo	5.1.0
Input Validation	express-validator	7.0.1
CORS	cors	2.8.5
IDE	Visual Studio Code	Latest
Version Control	Git	2.x
API Testing	Postman	Latest
Database GUI	MongoDB Compass	Latest
Deployment (Frontend)	Render Static Site	-
Deployment (Backend)	Render Web Service	-
Cloud Database	MongoDB Atlas	M0 Free Tier

3. ER DIAGRAM OF THE PROJECT

The ECO-SEVAKS database consists of three primary entities: **Users**, **Events**, and **Sessions**. The Entity-Relationship diagram below illustrates the relationships between these entities.





Relationships:

- **Users → Events (creates):** One user can create many events (1:N relationship)
- **Users ↔ Events (joins):** Many users can join many events (M:N relationship)
- **Users → Sessions (has):** One user has one active session (1:1 relationship)

4. SCHEMA OF THE PROJECT

Collection 1: users

Field	Data Type	Constraints	Description
-------	-----------	-------------	-------------

_id	ObjectId	Primary Key	Unique user identifier
name	String	Required, Min: 2 chars	User's full name
email	String	Required, Unique	Login credential
password	String	Required, Hashed	Bcrypt hashed (10 rounds)
role	String	Enum: 'user', 'admin'	Access control role
joinedEvents	Array[ObjectId]	FK → events	Events user registered for
createdEvents	Array[ObjectId]	FK → events	Events user organized
createdAt	Date	Default: now	Account creation timestamp

Indexes: email: 1 (unique)

Collection 2: events

Field	Data Type	Constraints	Description
_id	ObjectId	Primary Key	Unique event identifier
title	String	Required	Event name
description	String	Required	Detailed description
date	Date	Required	Event date and time
type	String	Enum: 'virtual', 'in-person'	Event format
location	String	Required	Venue or meeting link
tags	Array[String]	Optional	Categories (cleanup, plantation)
imageUrl	String	Optional	Event image URL
organizer	ObjectId	FK → users	Creator's user ID
attendees	Array[ObjectId]	FK → users	Registered volunteers
status	String	Enum: pending/approved/rejected/restricted	Approval status
maxVolunteers	Number	Default: 0 (unlimited)	Capacity limit
whatToBring	String	Optional	Instructions for volunteers
createdAt	Date	Default: now	Creation timestamp

Indexes: status: 1, date: 1, organizer: 1, {status: 1, date: 1} (compound)

Collection 3: sessions

Field	Data Type	Constraints	Description
_id	String	Primary Key	Session ID
session	Object	Contains cookie config	Session details
expires	Date	TTL Index	Auto-deletion time

TTL Index: expires: 1 with expireAfterSeconds: 0

5. USER INTERFACE DESIGN

The ECO-SEVAKS frontend is built using React.js with Tailwind CSS for styling. Below are screenshots of the key user interface pages with descriptions.

Figure 5.1: Home Page

[INSERT SCREENSHOT HERE]

Description: The home page displays all approved environmental events in a responsive grid layout. Users can browse events, filter by type (virtual/in-person), and click on event cards to view detailed information. The hero section showcases the platform's mission with call-to-action buttons.

Figure 5.2: Login Page

[INSERT SCREENSHOT HERE]

Description: The login page features a clean form with email and password fields using floating label inputs. Form validation provides real-time feedback, and toast notifications indicate success or error states.

Figure 5.3: Registration Page

[INSERT SCREENSHOT HERE]

Description: New users can create accounts by providing their name, email, and password. The form includes validation for email format and password strength, with confirmation matching.

Figure 5.4: User Dashboard

[INSERT SCREENSHOT HERE]

Description: The personalized dashboard displays user statistics with animated counters (Events Joined, Upcoming Events, Completed Events, Events Created). Quick action buttons allow users to create new events or browse existing ones. Sections show joined events and created events with status badges.

Figure 5.5: Event Detail Page

[INSERT SCREENSHOT HERE]

Description: The event detail page shows comprehensive information including event image, title, description, date/time, location, tags, maximum volunteers, and "What to Bring" instructions. Users can join or leave events with one click. Organizers can view and manage registered volunteers.

Figure 5.6: Create Event Page

[INSERT SCREENSHOT HERE]

Description: Event organizers can create new events using an intuitive form with fields for title, description, date, type selection, location, image URL (with live preview), tags, capacity, and volunteer instructions. Events are submitted for admin approval.

Figure 5.7: Admin Panel - Pending Events

[INSERT SCREENSHOT HERE]

Description: The admin panel provides a tabbed interface showing pending, approved, and restricted events with count badges. Admins can preview event details in a premium modal and approve or reject pending submissions.

Figure 5.8: Mobile Responsive View

[INSERT SCREENSHOT HERE]

Description: The application is fully responsive, adapting to mobile screens with optimized navigation, stacked layouts, and touch-friendly elements.

6. BACK END DESIGN

The ECO-SEVAKS backend is built using Node.js with Express.js framework, implementing a RESTful API architecture.

Backend Structure

```
backend/
  └── server.js      # Main entry point
  └── config/
    └── db.js        # MongoDB connection
  └── models/
    └── User.js      # User schema
    └── Event.js     # Event schema
  └── routes/
    └── auth.js       # Authentication routes
    └── events.js     # Event CRUD routes
    └── users.js      # User routes
  └── middleware/
    └── auth.js       # Auth middleware
  └── collections/
    └── users.js      # User DB operations
    └── events.js      # Event DB operations
```

API Endpoints

Authentication (/api/auth):

Method	Endpoint	Description
POST	/register	Create new user account
POST	/login	Authenticate and create session
POST	/logout	Destroy user session
GET	/me	Get current authenticated user

Events (/api/events):

Method	Endpoint	Description
--------	----------	-------------

GET	/	Get all approved events
GET	/pending	Get pending events (admin)
GET	/:id	Get single event details
POST	/	Create new event
PUT	/:id/approve	Approve event (admin)
PUT	/:id/reject	Reject event (admin)
PUT	/:id/restrict	Toggle event restriction
POST	/:id/join	RSVP to event
DELETE	/:id/leave	Cancel RSVP
DELETE	/:id	Delete event
GET	/:id/volunteers	Get volunteer list
DELETE	/:id/volunteers/:userId	Remove volunteer

Figure 6.1: API Testing with Postman

[INSERT SCREENSHOT HERE]

Description: Screenshot showing Postman API testing with sample requests and responses for authentication and event endpoints.

Figure 6.2: MongoDB Compass - Database Collections

[INSERT SCREENSHOT HERE]

Description: MongoDB Compass view showing the ecosevaks database with users, events, and sessions collections.

Figure 6.3: Server Console Output

[INSERT SCREENSHOT HERE]

Description: Terminal output showing successful server startup, MongoDB connection, and API request logs.

Security Implementation

- **Password Hashing:** bcrypt with 10 salt rounds
- **Session Management:** express-session with MongoDB store
- **CORS Configuration:** Configured for frontend origin with credentials
- **Input Validation:** express-validator on all endpoints
- **Protected Routes:** Authentication middleware for restricted endpoints
- **Role-Based Access:** Authorization middleware for admin-only routes

7. CONCLUSION AND FUTURE WORK

Conclusion

The development of **ECO-SEVAKS** has successfully demonstrated the application of modern full-stack web development technologies to address real-world environmental challenges. Through this project, we have created a comprehensive platform that connects environmental volunteers with eco-friendly events.

Key Achievements:

1. Technical Implementation:

- Built a complete full-stack application using React, Node.js, Express, and MongoDB
- Implemented secure session-based authentication with role-based access control
- Created a RESTful API with 15+ endpoints for all CRUD operations
- Designed a responsive UI using Tailwind CSS for all device sizes
- Successfully deployed the application to cloud services (Render, MongoDB Atlas)

2. Core Features Delivered:

- User authentication and authorization (user/admin roles)
- Event discovery with filtering and search
- Event creation with admin approval workflow
- RSVP management with capacity tracking
- Volunteer management for event organizers
- Personalized dashboards with statistics
- Admin panel for platform oversight

3. Learning Outcomes:

- Gained practical experience with React hooks, context API, and component architecture
- Mastered backend development with Express.js, middleware, and routing
- Learned MongoDB schema design and NoSQL data modeling
- Understood deployment practices and environment configuration
- Developed skills in responsive web design and modern UI/UX principles

Future Work

The following enhancements are planned for future versions:

1. Mobile Application: Develop native iOS and Android apps using React Native with push notifications

2. Advanced Features:

- Google Calendar integration for event sync
- Impact tracking (trees planted, waste collected)
- Gamification with volunteer badges and leaderboards
- Social media sharing integration

3. AI/ML Integration:

- Personalized event recommendations
- Chatbot for event discovery assistance
- Sentiment analysis on event reviews

4. Enhanced Communication:

- In-app messaging between organizers and volunteers

- Email and SMS notifications for event reminders
- Video conferencing for virtual events

5. Payment Integration:

- Donation support for environmental causes
- Paid workshop ticketing

6. Multi-language Support:

- Hindi, Tamil, Bengali, and other Indian language translations
-

8. REFERENCES

1. **React.js Documentation** - <https://react.dev/>
 2. **Node.js Documentation** - <https://nodejs.org/docs/>
 3. **Express.js Guide** - <https://expressjs.com/>
 4. **MongoDB Manual** - <https://www.mongodb.com/docs/manual/>
 5. **Tailwind CSS Documentation** - <https://tailwindcss.com/docs>
 6. **Vite Build Tool** - <https://vitejs.dev/>
 7. **Axios HTTP Client** - <https://axios-http.com/docs/>
 8. **bcryptjs Library** - <https://github.com/dcodeIO/bcrypt.js>
 9. **express-session** - <https://github.com/expressjs/session>
 10. **MDN Web Docs** - <https://developer.mozilla.org/>
 11. **Stack Overflow** - <https://stackoverflow.com/>
 12. **Render Deployment Docs** - <https://render.com/docs>
 13. **MongoDB Atlas Documentation** - <https://www.mongodb.com/docs/atlas/>
 14. **Lucide React Icons** - <https://lucide.dev/>
 15. **React Hot Toast** - <https://react-hot-toast.com/>
 16. **REST API Design Guide** - <https://restfulapi.net/>
 17. **OWASP Security Guidelines** - <https://owasp.org/>
-

END OF REPORT