**Some popular ticketing tools:**

1. \*\*Jira Service Management\*\*:

Popular among IT and software development teams for tracking issues and project management.

1. \*\*Zendesk\*\*:

A widely used customer service software with robust ticketing capabilities.

1. \*\*Freshdesk\*\*:

An easy-to-use helpdesk system with multichannel support.

1. \*\*ServiceNow\*\*:

An enterprise-level IT service management tool.

1. \*\*Help Scout\*\*:

Known for its simplicity and user-friendly interface.

1. \*\*Kayako\*\*:

Provides unified customer service across multiple channels.

1. \*\*SysAid\*\*:

Combines help desk and asset management.

1. \*\*Spiceworks\*\*:

A free IT help desk and network management tool.

1. \*\*osTicket\*\*:

An open-source ticketing system.

1. \*\*Zoho Desk\*\*:

Part of the Zoho suite, it offers multichannel support and automation.

Each tool has its own set of features and advantages, so the best choice depends on your specific needs and use cases.

**Open Source Ticketing Tools:**

Here are some popular open-source ticketing tools:

1. \*\*osTicket\*\*

- A widely used open-source support ticket system.

- Features include email piping, auto-responses, and customizable fields.

- [osTicket GitHub](https://github.com/osTicket/osTicket)

2. \*\*OTRS (Open Ticket Request System)\*\*

- A robust ticketing system for managing customer requests.

- Features include workflow automation, customizable templates, and extensive reporting.

- [OTRS GitHub](https://github.com/OTRS/otrs)

3. \*\*Zammad\*\*

- A web-based, open-source user support/ticketing solution.

- Features include chat integration, social media integration, and detailed reporting.

- [Zammad GitHub](https://github.com/zammad/zammad)

4. \*\*Request Tracker (RT)\*\*

- Developed by Best Practical Solutions, it's a powerful tool for tracking tasks and managing customer requests.

- Features include a robust email interface, configurable dashboards, and comprehensive search capabilities.

- [Request Tracker GitHub](https://github.com/bestpractical/rt)

5. \*\*Faveo Helpdesk\*\*

- An open-source ticketing system and support management tool.

- Features include SLA management, custom ticket fields, and reporting tools.

- [Faveo GitHub](https://github.com/ladybirdweb/faveo-helpdesk)

6. \*\*UVdesk\*\*

- A community-driven helpdesk ticketing system.

- Features include email management, knowledge base, and task management.

- [UVdesk GitHub](https://github.com/uvdesk/community-skeleton)

7. \*\*Helpy\*\*

- A modern helpdesk platform combining ticketing, knowledge base, and community features.

- Features include multi-language support, ticket automation, and reporting tools.

- [Helpy GitHub](https://github.com/helpyio/helpy)

These tools provide a range of functionalities to help manage customer support and internal IT requests.

**Creating own ticketing tool**

Yes, you can create your own ticketing tool! Building a ticketing system from scratch allows you to customize it to meet your specific needs. Here are the key steps and considerations to help you get started:

### 1. \*\*Define Requirements\*\*

- \*\*User Roles\*\*: Define roles such as Admin, Support Agent, and Customer.

- \*\*Ticket Attributes\*\*: Determine the attributes each ticket should have (e.g., ID, subject, description, status, priority, creation date, etc.).

- \*\*Workflows\*\*: Outline how tickets will be created, assigned, updated, and closed.

- \*\*Notifications\*\*: Specify how and when users will be notified about ticket updates.

- \*\*Reports\*\*: Define the types of reports you need (e.g., open tickets, response time, resolution time).

### 2. \*\*Choose Technology Stack\*\*

- \*\*Frontend\*\*: HTML, CSS, JavaScript (React, Angular, or Vue.js).

- \*\*Backend\*\*: Node.js, Python (Django/Flask), Ruby on Rails, PHP (Laravel), or Java (Spring Boot).

- \*\*Database\*\*: PostgreSQL, MySQL, MongoDB, or SQLite.

- \*\*Server\*\*: Use platforms like AWS, Azure, or DigitalOcean for hosting.

### 3. \*\*Set Up Project Structure\*\*

- \*\*Frontend and Backend\*\*: Separate concerns by creating different directories for frontend and backend.

- \*\*Database Schema\*\*: Design your database schema to handle users, roles, tickets, comments, and any other relevant data.

### 4. \*\*Develop Core Features\*\*

- \*\*User Authentication\*\*: Implement sign-up, login, and role-based access control.

- \*\*Ticket Management\*\*: Build CRUD operations (Create, Read, Update, Delete) for tickets.

- \*\*Commenting System\*\*: Allow users to add comments to tickets.

- \*\*Notification System\*\*: Implement email or in-app notifications for ticket updates.

- \*\*Dashboard\*\*: Create a dashboard for users to view their tickets and for admins to manage all tickets.

### 5. \*\*Implement Additional Features\*\*

- \*\*Search and Filters\*\*: Allow users to search and filter tickets by various attributes.

- \*\*File Attachments\*\*: Enable users to attach files to tickets.

- \*\*Analytics and Reporting\*\*: Provide tools for generating reports and visualizing ticket data.

- \*\*Integrations\*\*: Integrate with other tools like email clients, chat systems, or third-party APIs.

### 6. \*\*Testing\*\*

- \*\*Unit Tests\*\*: Write tests for individual components and functions.

- \*\*Integration Tests\*\*: Test interactions between different parts of the application.

- \*\*User Acceptance Testing\*\*: Ensure the application meets the requirements and is user-friendly.

### 7. \*\*Deployment\*\*

- \*\*Continuous Integration/Continuous Deployment (CI/CD)\*\*: Set up pipelines to automate testing and deployment.

- \*\*Monitoring and Logging\*\*: Implement monitoring and logging to track the application's performance and issues.

### 8. \*\*Documentation\*\*

- \*\*User Guides\*\*: Provide documentation for end-users on how to use the system.

- \*\*Developer Guides\*\*: Document the codebase to help future developers understand and contribute to the project.

### Example Tools and Libraries

- \*\*Frontend\*\*: React.js, Bootstrap, Axios

- \*\*Backend\*\*: Express.js (Node.js), Django (Python), Laravel (PHP)

- \*\*Database\*\*: Sequelize (ORM for Node.js), SQLAlchemy (Python ORM), Eloquent (Laravel ORM)

- \*\*Authentication\*\*: JWT (JSON Web Tokens), OAuth

- \*\*Notifications\*\*: Nodemailer (Node.js), Django Email (Python)

- \*\*Testing\*\*: Jest (JavaScript), PyTest (Python), PHPUnit (PHP)

- \*\*Deployment\*\*: Docker, Kubernetes, Jenkins

By following these steps and leveraging existing libraries and frameworks, you can build a robust and customized ticketing system tailored to your needs.

**List of Open Source Ticketing Tools:**

Here are some popular open-source ticketing tools:

1. \*\*osTicket\*\*

- A widely used open-source support ticket system.

- Features: email piping, auto-responses, customizable fields.

- [osTicket GitHub](https://github.com/osTicket/osTicket)

2. \*\*OTRS (Open Ticket Request System)\*\*

- A robust ticketing system for managing customer requests.

- Features: workflow automation, customizable templates, extensive reporting.

- [OTRS GitHub](https://github.com/OTRS/otrs)

3. \*\*Zammad\*\*

- A web-based, open-source user support/ticketing solution.

- Features: chat integration, social media integration, detailed reporting.

- [Zammad GitHub](https://github.com/zammad/zammad)

4. \*\*Request Tracker (RT)\*\*

- Developed by Best Practical Solutions, it's a powerful tool for tracking tasks and managing customer requests.

- Features: robust email interface, configurable dashboards, comprehensive search capabilities.

- [Request Tracker GitHub](https://github.com/bestpractical/rt)

5. \*\*Faveo Helpdesk\*\*

- An open-source ticketing system and support management tool.

- Features: SLA management, custom ticket fields, reporting tools.

- [Faveo GitHub](https://github.com/ladybirdweb/faveo-helpdesk)

6. \*\*UVdesk\*\*

- A community-driven helpdesk ticketing system.

- Features: email management, knowledge base, task management.

- [UVdesk GitHub](https://github.com/uvdesk/community-skeleton)

7. \*\*Helpy\*\*

- A modern helpdesk platform combining ticketing, knowledge base, and community features.

- Features: multi-language support, ticket automation, reporting tools.

- [Helpy GitHub](https://github.com/helpyio/helpy)

8. \*\*GLPI\*\*

- An open-source IT asset management and ticketing tool.

- Features: inventory management, ticket tracking, reporting.

- [GLPI GitHub](https://github.com/glpi-project/glpi)

9. \*\*Trudesk\*\*

- An open-source help desk solution.

- Features: real-time updates, customizable workflows, reporting.

- [Trudesk GitHub](https://github.com/polonel/trudesk)

10. \*\*SuiteCRM\*\*

- An open-source CRM platform with a built-in ticketing system.

- Features: case management, workflow automation, reporting.

- [SuiteCRM GitHub](https://github.com/salesagility/SuiteCRM)

These tools offer various features to manage customer support and IT service requests, providing flexibility for different use cases and requirements.