**🧩 Software Requirements Specification (SRS)**

**Project: ORT Preparation Platform**

**1.0 Introduction**

**1.1 Purpose**

The purpose of this system is to create a **centralized, community-driven platform** for preparation for the **National Standardized Test (ORT)** in Kyrgyzstan.  
The platform aims to connect students preparing for ORT with educational materials, practice tests, and a moderation system that ensures quality content.

**1.2 Scope (MVP)**

The **Minimum Viable Product (MVP)** version of the platform will include:

* User registration and authentication.
* Creation and publication of learning articles (subject to moderation).
* Creation and publication of test questions (subject to moderation).
* Ability for users to take tests and view their results.

**1.3 Definitions and Abbreviations**

| **Term** | **Definition** |
| --- | --- |
| **ORT** | *Общереспубликанский тест (National Standardized Test)* — a national academic assessment in Kyrgyzstan. |
| **Content** | Any user-generated item (article, test question, etc.). |
| **Sandbox** | Queue of content waiting for moderation. |

**1.4 Target Audience (Roles)**

| **Role** | **Description** |
| --- | --- |
| **Guest** | Unregistered user. Can only view approved articles. |
| **User** | Registered participant (student). Can read articles, take tests, and propose new content. |
| **Moderator** | Trusted user with rights to manage and approve/reject content. |
| **Administrator** | System owner (you) with full system privileges, including management of moderators and sections. |

**2.0 Overall Description**

**2.1 Role and Permissions Overview**

| **Role** | **Permissions** |
| --- | --- |
| **Guest** | Can browse articles only. Cannot take tests, comment, or submit content. |
| **User** | Can read articles, take tests, view statistics, and submit new content (articles/questions) for moderation. |
| **Moderator** | All User permissions + access to the **Sandbox** for approving/rejecting content. |
| **Administrator** | All Moderator permissions + ability to assign/remove moderators and manage site sections. |

**2.2 System Architecture (Assumed)**

The system will be built using a **microservice architecture** to ensure scalability and flexibility.  
Planned services include:

* **Auth-Service** — handles user registration, login, and authentication (JWT).
* **Content-Service** — manages articles and test questions.
* **Moderation-Service** — handles content review, approval, and rejection logic.
* **Testing-Service** — manages test sessions, question randomization, and scoring logic.

**2.3 Technology Stack (Assumed)**

| **Layer** | **Technology** |
| --- | --- |
| **Backend** | Java (Spring Boot / Kotlin optional) |
| **Frontend** | Vue.js *(React as optional alternative)* |
| **Database** | PostgreSQL (for relational data) + possible MongoDB (for flexible content storage) |
| **Auth** | JWT-based authentication |
| **Deployment** | Dockerized microservices, orchestrated via Docker Compose / Kubernetes *(future expansion)* |
| **Version Control** | Git (GitHub or GitLab) |
| **CI/CD** | GitHub Actions or Jenkins *(future integration)* |

**2.4 System Objectives**

* Provide students with high-quality educational resources for ORT preparation.
* Encourage community participation through article and question submissions.
* Maintain content integrity via structured moderation workflows.
* Enable scalable architecture for potential national-level usage.

**3.0 Functional Requirements (User Stories)**

**3.1 Authentication Module**

| **ID** | **User Story** |
| --- | --- |
| **FR-1.1** | As a **Guest**, I want to register using email and password so that I can become a **User**. |
| **FR-1.2** | As a **User/Moderator/Admin**, I want to log in using my credentials to access the system. |
| **FR-1.3** | As a **Guest**, when trying to take a test, I should be prompted to register or log in first. |

**3.2 Content Module (Articles & Tests)**

| **ID** | **User Story** |
| --- | --- |
| **FR-2.1** | As a **User**, I want to create a draft article or question in my profile. |
| **FR-2.2** | As a **User**, I want to submit my draft for moderation. |
| **FR-2.3** | As a **Guest**, I want to read only approved (published) articles. |
| **FR-2.4** | As a **User**, I want to see test questions attached to an article. |

**3.3 Testing Module**

| **ID** | **User Story** |
| --- | --- |
| **FR-3.1** | As a **User**, I want to start a test linked to an article. |
| **FR-3.2** | As a **User**, I want to answer test questions (single/multiple choice). |
| **FR-3.3** | As a **User**, I want to finish a test and immediately see my score (X out of Y correct). |

**3.4 Moderation Module**

| **ID** | **User Story** |
| --- | --- |
| **FR-4.1** | As a **Moderator**, I want to view the “Sandbox” containing all pending content. |
| **FR-4.2** | As a **Moderator**, I want to open and review a submission, then click **Approve** (status → Approved). |
| **FR-4.3** | As a **Moderator**, I want to **Reject** a submission with a required “Reason” field (status → Needs Revision). |
| **FR-4.4** | As a **User**, I want to receive a notification if my content is rejected, along with the reason. |

**4.0 Non-Functional Requirements**

| **Category** | **Requirement** |
| --- | --- |
| **Performance** | Page load time should not exceed 3 seconds. |
| **Security** | Passwords must be securely hashed (e.g., bcrypt). |
| **Scalability** | The microservice architecture must allow independent scaling (e.g., during ORT test sessions). |
| **Compatibility** | The platform must function correctly in the latest versions of Chrome, Firefox, and Safari. |
| **Reliability** | The system should handle concurrent users without data loss or corruption. |
| **Usability** | The UI should be responsive and accessible on desktop and mobile devices. |

**5.0 Future Enhancements (Post-MVP)**

* Integration with online payment gateways (for tutor marketplace expansion).
* User ranking and gamification (XP, achievements).
* Real-time chat or comment system.
* AI-powered question recommendations and difficulty adjustment.
* Full ORT simulation mode.

**6.0 Project Deliverables**

* RESTful API documentation (Swagger / OpenAPI).
* Frontend prototype (Vue 3 + TailwindCSS).
* Database schema (ER diagram).
* Deployment plan (Dockerized microservices).
* Presentation slides for stakeholders (based on this SRS).