

CSCI 409– Senior Project

Bibarys Mukhambetiyar	202078886
Aibek Narik	202066142
Ibrahim Bekturgan	202197378
Yergaliyev Aron	202155785

Final Report

APRIL Robots Online App: Curriculum Management, GPT Automation, and Platform Roles

1. Executive Summary

The APRIL Robots Online App is a cloud solution that will help the treatment centers of the autism spectrum disorder in Kazakhstan. It is a pragmatic matter: data about therapy, including curriculum resources, session data, parent communication, and center knowledge, is disaggregated when on various files, spreadsheets, paper records, and informal communications.

Design & Implementation

It is a platform with an administrator role, specialist role and parent role, centralized curriculum management, multiple languages (Kazakh, Russian, English), integration of offline and cloud infrastructure and automatic deployment infrastructure.

It is composed of FastAPI (backend), Svelte/TypeScript (frontend), MongoDB (database), Docker/k3s (deployment), and GitHub Actions (CI/CD). Authenticating and controlling access and isolating permissions across centers are done using keycloak, JWT, RBAC and CORS.

Execution

The team accomplished 60+ GitHub issues across five work streams in Spring 2026; data sync and backend, user interface and interaction, infrastructure and deployment, authentication and security, and database schema integration. The work was done with the coordination meetings after every two weeks.

Results & Evaluation

The platform has centralized curriculum, role specific dashboards, upstream synchronization, and automated k3s deployment.

Technical performance: Load test (2 minutes, 10,855 requests) got 90.56 requests/second with a single failure (0.009% failure rate) and a great response time (p50: 3ms, p99: 3ms).

User comments: Experts (n=6, avg 8.6/10) cited user friendly interfaces and good synchronization. Parents (n=4, avg 8.1/10) appreciated access to progress in therapy. These results are preliminary signs of technical feasibility and usefulness- not clinical validation.

Scope & Limitations

The platform does not assist in clinical decision-making but highly organized digital assistance in therapy administration. A lesson-plan / script assistant based on Google Gemini (gemini-2.5-flash) was prototyped with the help of the LLM. The module exists in the codebase but its router is turned off in production as priorities shifted to core functionality and deployment stability. The role-based frontend is partially implemented: the administrator interface is

complete, while the specialist and parent interfaces are in active design review and currently reuse the administrator layout as a template, pending role-specific finalization. The next-generation work must include further validation of more users, testing synchronization in longer real-world scenarios, and formal workflow of specialist approval of curriculum content.

2. Introduction

The treatment of autism spectrum disorder relies on organized, repetitive and personalized activities. Experts design learning resources, customize assignments to personal needs, document progress, and discuss with parents. Once these workflows take place in more than one center, fragmentation becomes an issue: the information that is distributed across spreadsheets, local documents, paper notes, and informal messaging is hard to synchronize, search, and access uniformly.

The Problem Context

The number of children with ASD in need of intensive care in Kazakhstan is estimated to be 14500. Numerous therapy facilities have independent data systems that cannot be long-term tracked, allowing parental participation, and cross-facility knowledge. To have a digital platform in this context, a number of practical requirements have to be met:

- Provide access to multiple centers with different user permissions.
- Keep confidential child and therapy data.
- Support various user groups (administrators, specialists, parents) with different requirements and workflows.
- Encourage multi lingualism (Kazakh, Russian, English)

Platform Design

The APRIL Robots Online App is a solution to these needs as it offers role-based interfaces and central data management. Data and user privileges are managed by administrators. Curriculum resources, child profiles, therapy skills, and session documentation can be viewed by experts. The progress of their child is also regulated to parents without disclosing information about the administrative or internal specialists.

The platform offers organised online assistance with the administration of therapies and coordination - it is not a substitute of professional judgment or clinical decision-making.

Significance

The project uses the methods of computing and web infrastructure to a real educational and therapeutic issue. The use of a common platform can reduce the disintegration of records, promote transparency with parents, provide consistency in the management of the curriculum, and provide a data foundation to carry out future research on robot-assisted therapy (Amirova et

al., 2023). As the system will be working with sensitive child data, and processes associated with the therapy, it will require a careful consideration of technical stability, usability and information security.

Report Structure

This report covers the project in terms of identification of the problem, implementation and evaluation. Section 3 discusses constraints. Section 4 discusses related work. Section 5 describes the role-based and architecture. Section 6 outlines implementation in five streams of work. Section 7 shows technical and user-based evaluation outcomes. Section 8 sums up and provides future work.

3. Identification of Constraints

The project was conditioned by a number of limitations that are inherent to a sensitive environment based on therapy, but not a web application.

Economic

Non-profit and therapy centers have low budgets. The project was based on open-source technologies (FastAPI, Svelte, MongoDB, Docker, k3s, GitHub Actions) to minimize the cost of licensing and allow the project to be maintained by the institutions in the future. Nevertheless, server infrastructure, backups, monitoring, technical support, and staff training are nevertheless needed in real deployment.

Infrastructure

Unstable internet connectivity among locations necessitated offline synchronization to the cloud. This enables transfer of therapy information between disconnected center environments to the common platform. Multi-center operation requires synchronization, which complicates matters, especially when there are multiple centers updating data at the same time, but this is necessary to achieve multi-center operation.

Social & Cultural

The users consist of administrators, specialists, and parents who have varying technical expectations. Adoption required multilingual assistance (Kazakh, Russian, English) because users will work in their familiar language based on their position and place.

Legal & Data Governance

The data localization requirement in Kazakhstan required it to be deployed in Kazakhstan. It is implemented on a rented vps infrastructure using k3s, keeping the data in the country. Data security is implemented with the help of MongoDB access controls, role-based access control (RBAC), JWT authentication, Keycloak integration, encrypted communication, activity logging, and data export/deletion features. These security controls guard application and infrastructure level sensitive child and therapy data.

Ethical & Safety

The platform manages the data of children with ASD, which is why privacy, consent, and role-based access is critical, as well as responsible record-handling. False documentation, unauthorized access, or inappropriate content may influence the planning of the therapy and the trust of the family. These risks are addressed in the system with the help of JWT authentication, RBAC, integration with Keycloak, secure API routes, encrypted communication, logging of activities, and the ability to export and delete data.

Manufacturability & Sustainability

The project deployed using Docker, k3s, GitHub Actions, and documented deployment processes to support reproducible deployment and maintenance. Sustainability in the long run would involve defined organizational processes of user management, curriculum changes, and support requests as well as further stakeholder testing and commitment of ownership.

Trade-offs

- Synchronization: Allows access to shared locations across centers but enhances security and complexity of deployment.
- Offline operation: Supports real center conditions, but further complicates synchronization.
- Multilingual support: Enhances accessibility but needs to control the quality of translations.

4. Background & Related Work

ASD Therapy Context

The common activities in ASD support include systematic activities, practice, monitoring of progress, and collaboration between specialists and families. These workflows may be assisted

with technology aimed at supporting professional practice as opposed to substituting clinical judgment.

Robot-Assisted Therapy Research

Social robots (NAO, Furhat) have been studied in the previous work in ASD therapy (Indian Express, 2023). These robots can offer predictable interaction, facilitate interaction, and generate organized gathering of behavioral information (Diehl et al., 2012; Rakhymbayeva et al., 2021). This work has been developed by the Nazarbayev University research program via the research on long-term and adaptive robot-assisted therapy (Nazarbayev University, 2024).

APRIL Platform Role

The APRIL Robots Online App facilitates a complementary layer: infrastructure that enables the delivery of therapy across multiple centers. Although the main idea of robot-assisted therapy is to interact with a robot when performing activities, the surrounding ecosystem is controlled by this platform, such as curriculum organization, session data, user roles, parent visibility, offline-to-cloud synchronization, and cross-center coordination. It is an aid and not a substitute to therapy or robotics research.

Practical Alternatives

The disconnected methods prevalent in therapy centers are paper records (easy to search but hard to coordinate), spreadsheets (weak with many users and roles), messaging (not formal, no long-term structure), or generic learning (no designed therapy workflows). Both of them pose a hindrance to uniform curriculum management, parent involvement, and inter-center data sharing.

Technical Approach

The platform is an integration of role-based web design, authentication/authorization, offline-to-cloud synchronization and infrastructure automation. FastAPI and Svelte (responsive web platform), MongoDB (flexible therapy data), Keycloak/JWT/RBAC/CORS (authentication and access control), Docker/k3s/GitHub Actions (deployment automation) are all included in the stack.

Distinctive Contribution

The value of the project is in combining data management of therapy, curriculum organization, parent access, synchronization, multilingual interfaces and deployment infrastructure into one platform which is specific to multi-center workflows in ASD therapy - something that generic alternatives fail to do.

5. Project Approach

APRIL Robots Online App is a web platform that was created as a five layer app. The frontend layer is developed based on Svelte, TypeScript, and responsive design to deliver role-based interfaces. FastAPI, pydantic validation and access to MongoDB are used in the backend API to handle application logic and data. MongoDB collections are utilized in the database layer to store therapy data, curriculum content, users, roles, centers, and records associated with synchronization. The synchronization layer deals with offline/cloud data transfer, access using JWT, and conflict resolution. GitHub Actions, Docker and k3s are used in the DevOps layer to automate the deployment.

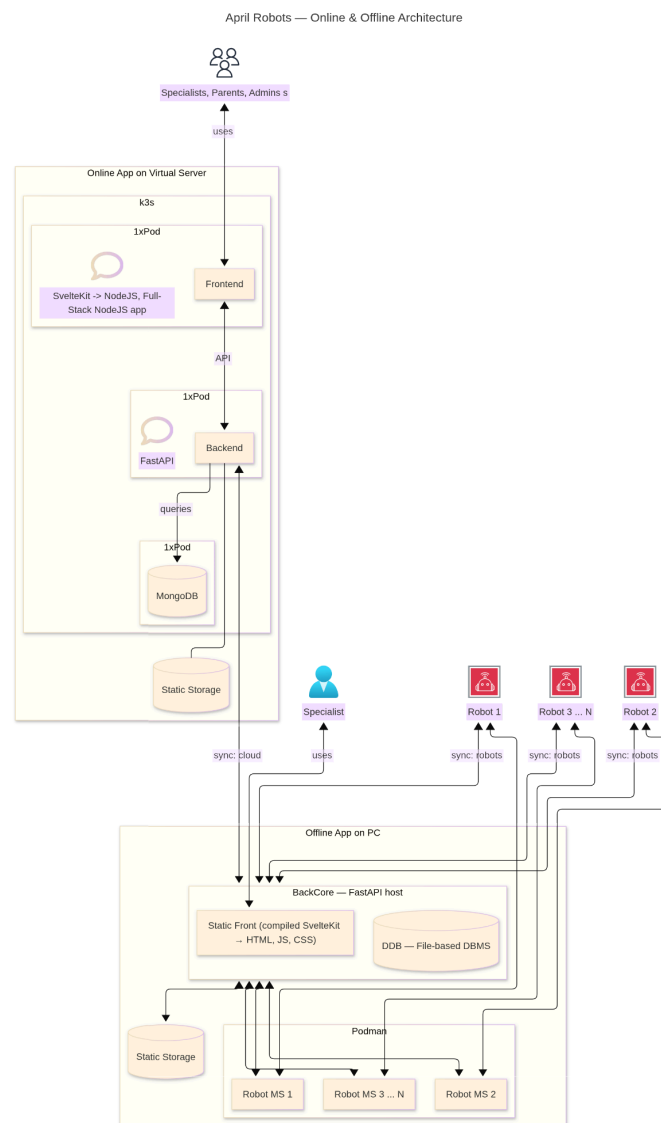


Figure 1. Current Architectural Model (combines online and offline architecture)

Role-Based Design

The role-based design is central to the system. Access control, authentication, and user role separation (administrator, specialist, parent) are implemented end-to-end on the backend through RBAC, JWT, and Keycloak integration. On the frontend, the administrator interface is fully implemented and supports center management, user permissions, and system organization. The specialist and parent interfaces are currently in active design review with the project supervisor: their role-scoped layouts are in place and reuse the administrator interface as a scaffold, but role-specific data filtering, write permissions, and final page designs are pending and are planned for the next development iteration. A public landing page is also implemented and provides center information and an interactive map of therapy locations.

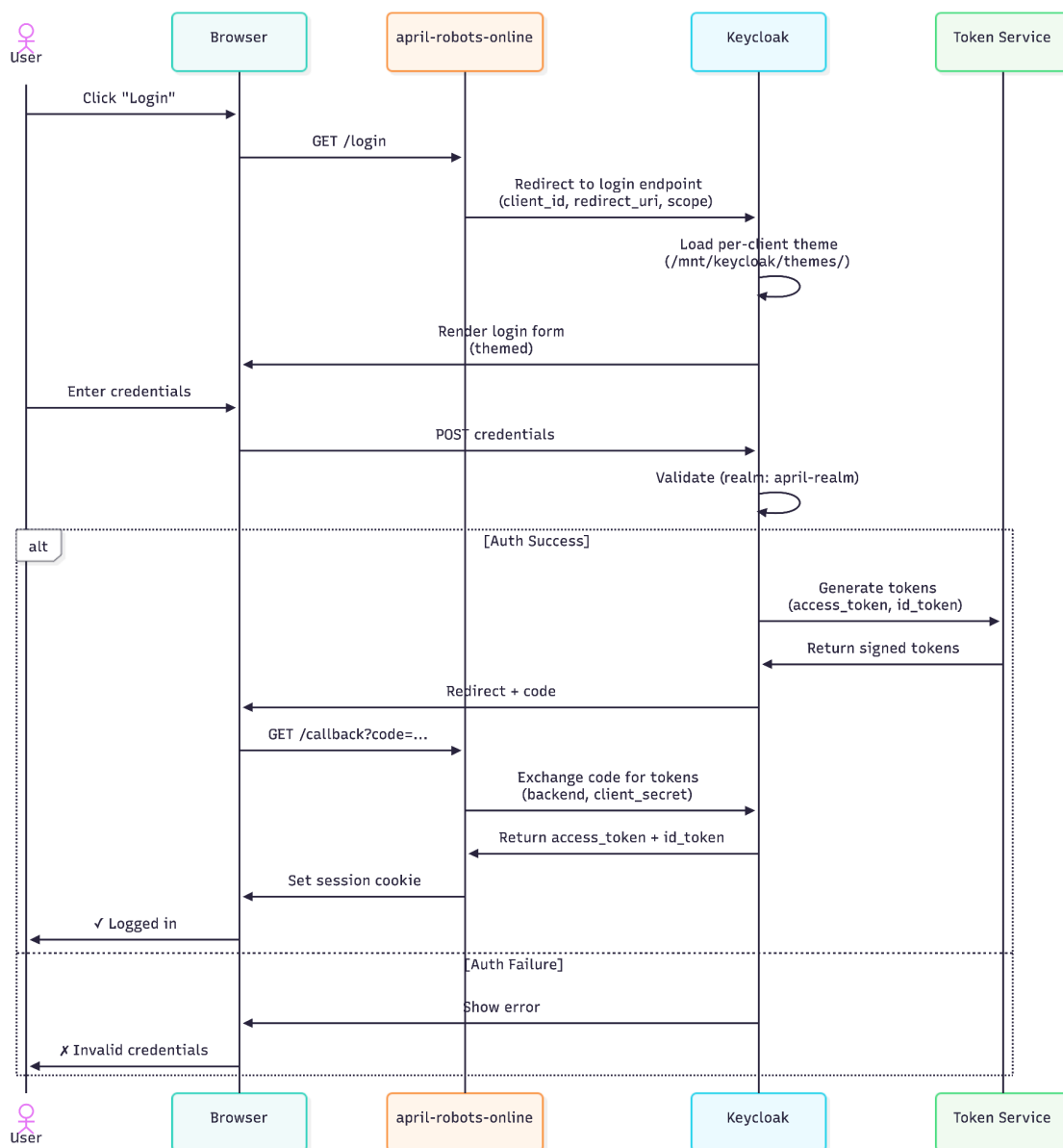


Figure 2. Login and Role-Based Access

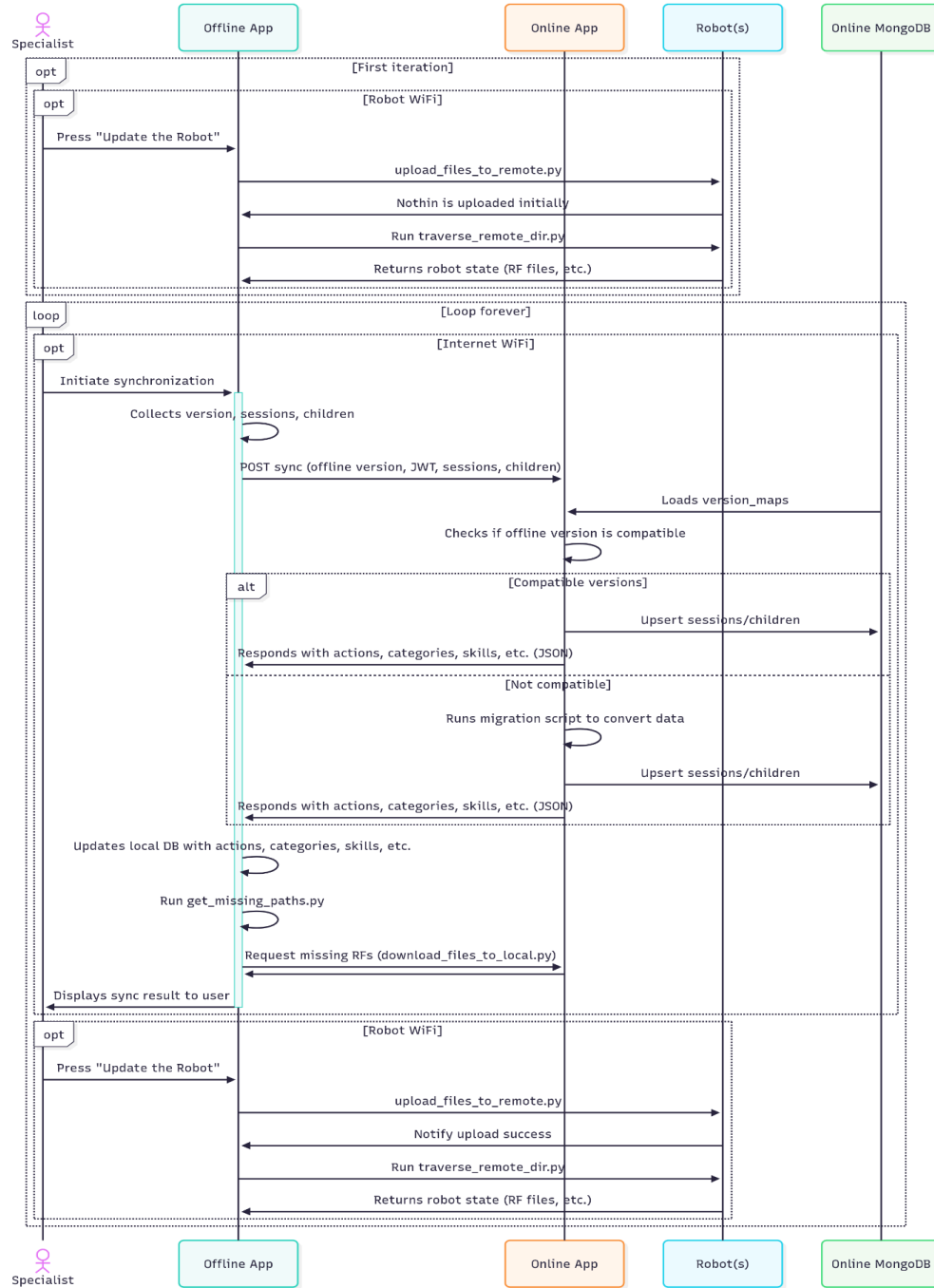


Figure 3. Interaction between Online and Offline applications

Curriculum Management

The curriculum management was aimed at minimizing reliance on documents and spreadsheets that were not interconnected. Curriculum data was imported into the spreadsheet and the formats of older data were transformed into the present schema. The database schema was restructured to facilitate synchronization, multi-center queries, parent-child associations, specialist-center assignments and compliance with ABLLS-R therapeutic standards. The platform consists of therapy skills, specialist grades, categories, subskills, evaluation grades, metrics, and questions. The activities in the curriculum and robot-specific metrics are scoped to the two robot platforms that the partner centers, NAO and Furhat, operate on, such that categories, actions, and metric settings can be filtered and written according to the type of robot.

Synchronization

Synchronization was one of the most important technical workflows. The team finished the offline synchronization to cloud pipeline and the upstream synchronization module. The system assists with transfer of session data, integrity control, conversion of older formats and debugging, audit trails and compliance related activity trace logging. A significant problem was synchronization race conditions: corruption of the data took place when more than two centers tried to synchronize simultaneously. The team tackled this by the use of transaction boundaries and version tracking.

Authentication & Security

Through a number of integrated components, authentication and security were addressed. JWT authentication is used to separate access by center, secured routes limit API access, Keycloak offers single sign-on and identity management to multi-center specialists, and CORS configuration facilitates secure communication between frontend and backend services. The parts were not made, but assembled.

User Interface & Frontend State Management

The project involved a lot of redesigning of the user-interface. The team redesigned navigation, sidebar, categories, role-based dashboards, authentication screen, registration, password recovery, profile page, skills page, center management, curriculum page and multilingual localization. With the expansion of the frontend, state management became a viable issue. As the interface complexity grew, prop drilling and deep component hierarchies became prevalent. Reactive class-based state has replaced prop drilling with reactive class-based state (e.g.

AuthState in src/lib/auth/auth.svelte.ts) with Svelte 5 runes, as state and derived (e.g. \$state, \$derived). This minimized complexity in the frontend and enabled quicker feature development.

```
JSON
# app/users/constants.py
class UserRoleEnum(str, Enum):
    ADMIN = "admin"
    PARENT = "parent"
    SPECIALIST = "specialist"

# app/auth/dependencies.py
def permit_to(roles: set[UserRoleEnum]):
    def permission_dependency(user: UserDep):
        if user.role not in roles:
            raise HTTPException(status.HTTP_403_FORBIDDEN)
        return user
    return Depends(permission_dependency)

AdminDep = Annotated[User, permit_to({UserRoleEnum.ADMIN})]
```

Listing 1. Role enum and the FastAPI dependency used to enforce role-based access on protected routes.

Deployment Infrastructure

GitHub Actions, Docker, and k3s were used to implement deployment. Tests are run on pull requests automatically, Docker images are built and pushed to GHCR on pushes to staging (→ staging env) and release/ (→ prod) and deployed with kubectl rollout restart statefulset via SSH. The infrastructure setup also comprised k3s CRD setup of the backend and frontend services, enduring storage in volumes and bind mounts, deployment/API/operational documentation.

```
JSON
on:
  push:
    branches: ["develop", "staging", "release/**"]
```

Listing 2. CI/CD concrete evidence

6. Project Execution

The Spring 2026 work aimed at transferring the APRIL Robots Online App off a series of implementation tasks and on to a more integrated platform. The team was able to finish over 60

GitHub issues which were categorized into five large work streams: data synchronization and backend, user interface and interaction, infrastructure and deployment, authentication and security, and database schema and integration. These streams were created simultaneously with bi-weekly synchronization meetings to ensure the team members are aligned.

Data Synchronization & Backend

The data synchronization and backend stream solved one of the most critical operational issues: transfer of data of therapy and sessions between offline centers and the cloud platform. The team finished with the offline-to-cloud synchronization pipeline and upstream synchronization. This involved transfer of session data, integrity checks, compatibility with older data formats and full logging of debugging and auditing. One of the major issues was the situation where several centers tried to synchronize concurrently, thus posing a risk of corruption of data. To mitigate the possibility of race conditions, the team added transaction boundaries and version tracking.

User Interface & Interaction

The user-interface stream was concerned with the usability of the system by various roles. The team re-architected the platform on three primary role-based interfaces, which are administrator, specialist, and parent. Sidebars, categories and better action hierarchy were used to restructure navigation. Authentication, registration and password recovery screens were introduced or enhanced. For managing centers and permissions, adm pages were developed. Additional elements were added including profile and skills pages and curriculum management pages including therapy activities and resources. The opening page has also been created with information about the public center and interactive map.

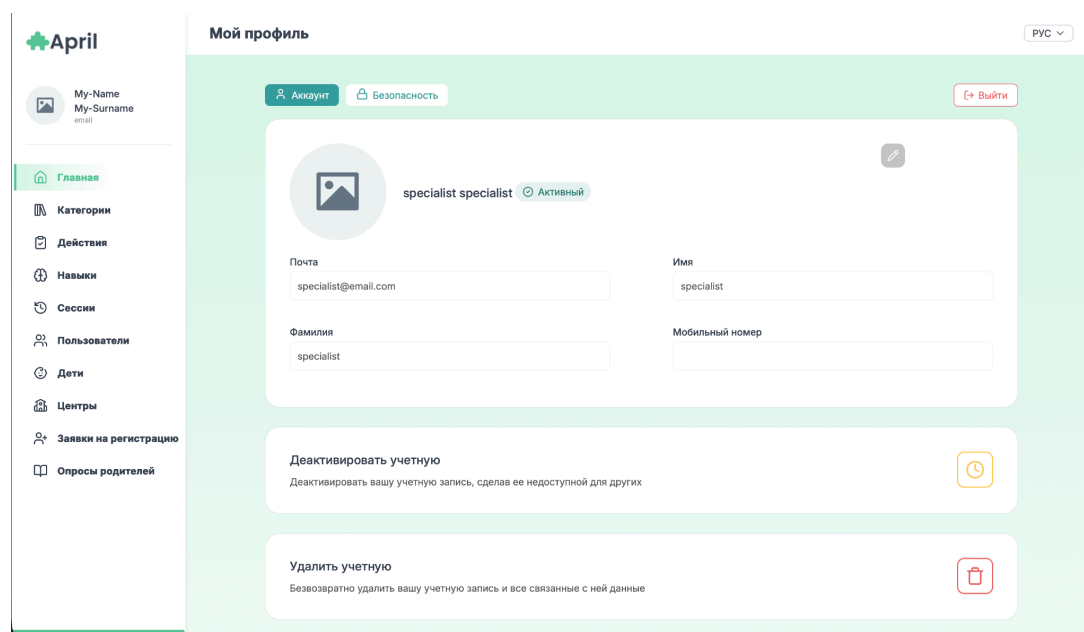


Image 1. Specialist View: Profile

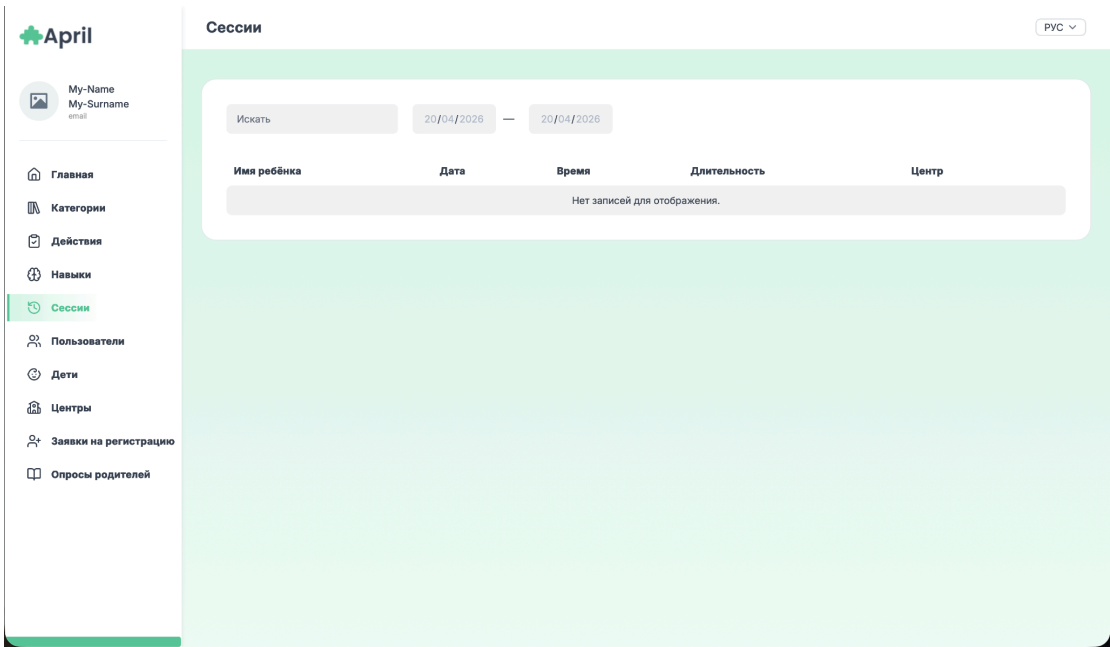


Image 2. Specialist View: Sessions

Infrastructure & Deployment

The automated build-and-deploy environment was created by the infrastructure and deployment stream. Both the backend and frontend repositories were set up using GitHub Actions workflows, and each of the pipelines was defined as either development, staging, or production build based on the Git branch. During each run, the workflow will install dependencies, compile the project and, when the test suites are complete, will automatically run tests prior to packaging. To stage and release builds, Docker images are pushed to the GitHub Container Registry and deployed to the k3s cluster, which runs the backend and frontend as separate services. The release builds also capture compatibility data between the cloud platform and the offline clients, and this way ensure that synchronization is maintained across the versions. Besides the pipeline, the team was ready to deploy, document API, and operational, and perform WebApp QA and acceptance testing with partner centers.

Monitoring & Observability

Prometheus and Grafana are deployed as pods in the k3s cluster. The Prometheus client is instrumented on the backend application (app/metrics/prometheus.py, MetricsMiddleware) and it has an endpoint, accessible to the administrator only (GET /prometheus), which is scraped by Prometheus, and dashboards are set up in the Grafana instance. Nevertheless, metrics gathering and dashboards were not properly configured to be used in production. At present, the infrastructure accommodates the implementation of these tools but needs further adjustment to have meaningful metrics collection, alerting policies, and operational dashboards. This output is not implemented to the current deployment stages until the production monitoring needs are determined.

Authentication & Security

The stream of authentication and security was needed as the platform contains sensitive child and therapy-related information. The team deployed a per-center JWT authentication, secured API routes, Keycloak SSO and identity management, and CORS configuration to ensure secure frontend-backend communication. These resolutions favored RBAC and assisted in isolating access among users and centers.

Database Schema & Integration

The database schema and integration stream were used to discuss the changes in the therapy data structures. The team reorganized the MongoDB collections to support the synchronization and multi-centre queries. Curriculum data in spreadsheet format were imported, and the old formats converted. Assignments that were specialist-centre based and those based on parent-child linking were redesigned. ABLLS-R therapeutic standards were in line with skills hierarchy and specialist grades. Other bugs that the team resolved were serialization bugs, duplicate ID bugs and registration validation bugs.

Overall, implementing the project proved that no one feature existed but the workflow integration was the main challenge. Both of them affected each other: synchronization, authentication, schema design, UI state and deployment. Issue-based workflow and bi-weekly meetings assisted the group in organizing these dependencies and keep parallel development.

7. Evaluation

The test involved technical behavior and acceptance by the initial user. The evidence at hand suggests a tentative conclusion: the platform is technically feasible and early usable, but has not been proven to be a long-term clinical or institutional system.

Load Test Results

Test Configuration

- Duration: 2 minutes
- Total requests: 10,855
- Throughput: 90.56 requests/second
- Failed requests: 1 (0.009% error rate)

Response Time Metrics

- Minimum: 1 ms
- Average: 14.7 ms
- p50 (median): 3 ms
- p90: 8 ms
- p95: 4 ms
- p99: 3 ms
- Maximum: 1,148 ms

Assessment: Excellent response times across all percentiles. The system handles requests quickly with minimal latency.

Endpoints Tested

- me: 14.6%
- childrenList: 12.2%
- sessionsList: 12.2%
- actionsList: 9.8%
- categoriesList: 9.8%
- skillsList: 9.8%
- subSkillsList: 9.8%
- evaluationGrades: 4.9%
- evaluationMetrics: 4.9%
- evaluationQuestions: 4.9%
- publicCentersPreview: 4.9%

- syncEmptyPayload: 2.4%

Failures

- 1 RemoteDisconnected error on GET /centers/preview (likely transient network issue)

Assessment: 0.009% error rate is excellent. Single failure among 10,855 requests indicates a robust system.

The test involved technical behavior and acceptance by the initial user. The evidence at hand suggests a tentative conclusion: the platform is technically feasible and early usable, but has not been proven to be a long-term clinical or institutional system.

User Satisfaction

The original report contains preliminary user-satisfaction feedback of experts and parents:

- Experts: n=6, mean satisfaction 8.6/10.
- Parents: n=4, average satisfaction 8.1/10

Experts were reported to be intuitive to the interface, the synchronization to be reliable, and they said that they would use the system on a daily basis. Parents were reportedly thankful to be able to see more into the progress of therapy and to be more engaged. These are positive findings, although the sample is small. These results should not be used to declare that the report was widely adopted or clinically effective.

Acceptance Testing and Problem validation.

QA testing and acceptance testing were done with partner centers: Asyl Miras, DRC Kids, and Bereke. This enhances the assessment since the platform was not evaluated as a standalone classroom prototype.

The platform seems to minimize fragmentation as it offers a common source of session data. It enhances parent visibility through provision of the parents with selected progress information. It minimizes technical obstacles through enabling curriculum management devoid of programming. It provides offline-to-cloud operation and synchronization to offline challenges. These results confirm the computing-based solution on an early platform level. Nonetheless, there is still a need to have stronger evidence on the long-term use, therapy effects, synchronization in the real-life conditions in the long-term.

8. Conclusion and possible future work

Conclusion & Future Work

The APRIL Robots Online App is created to assist the ASD treatment centers in Kazakhstan with the help of integrating curriculum management, role-based dashboards, multilingual interfaces, synchronization, authentication, and deploying automation in a single platform. The problem that was tackled by the project was as follows: information regarding therapy may get disjointed when centers use different local files, spreadsheets, paper records, and informal communication channels.

The project provided over 60 implementation activities in five work streams and created a functional system that was deployed using GitHub Actions, Docker, and k3s. The system has administrator, specialist, and parent interfaces, Kazakh, Russian, and English localization, upstream synchronization, Keycloak/JWT/RBAC authentication, and MongoDB schema changes of curriculum, session, child, parent, specialist, and center-related data. Partner therapy delivery locations were also involved in the project: Asyl Miras, DRC Kids, and Bereke.

The results of the evaluation were positive but preliminary. The load test revealed that there was high throughput and only one failure out of 10,855 requests. Feedbacks on user-satisfaction by six specialists and four parents were also positive. These findings justify the conclusion that the platform is technically possible and comprehensible to the first users. They are yet not demonstrations of long-term operational readiness or clinical efficacy.

Future Work

1. Expand evaluation to incorporate more specialists, parents, and administrators in partner centers.
2. Test concurrency to more realistic, longer network conditions.
3. Put in place official procedures of specialist endorsement of curriculum material.
4. Include backup, monitoring, incident-response and maintenance procedures to the deployment process.
5. Test multilingual text with non-experts and professionals in the field to make it understandable and coherent.
6. Increase longitudinal analytics, sensitive action audit logs, and enhanced parent-facing summaries, and strengthen collaborations with robot-assisted therapy research (APRIL).

Altogether, the project offers a solid technical and organizational base to a digital platform that can facilitate the management of curriculum, parent-involvement, and cross-centre coordination

in ASD therapy settings. Its current contribution can be defined as an operational and tested senior-project platform, whose initial results are encouraging and there are definite prospects to further its validations.

9. References

- Amirova, A., Telisheva, Z., Zhanatkyzy, A., & Sandygulova, A. (2023). Adaptive and long-term robot-assisted autism therapy for children with ASC: A reinforcement learning system. In Proceedings of the HRI '23 Companion.
https://leap-hri.github.io/2023/papers/LEAP-HRI_2023_paper_2946.pdf
- Diehl, J. J., Schmitt, L. M., Villano, M., & Crowell, C. R. (2012). Robot applications for autism: A comprehensive review. *Autism Research*, 13(4), 613-629.
<https://pubmed.ncbi.nlm.nih.gov/32706602/>
- Indian Express. (2023, December 23). Meet NAO, the social robot spearheading a new wave of autism treatment in Kazakhstan.
<https://indianexpress.com/article/technology/tech-news-technology/nao-robot-for-autism-anara-sandygulova-8986897/>
- Nazarbayev University. (2024). Research and development of AI-based robot-assisted therapy for children with autism.
<https://research.nu.edu.kz/en/projects/research-and-development-of-ai-based-robot-assisted-therapy-for-c>
- Rakhymbayeva, N., Seitzkazina, A., Turabayev, A., Pak, Y., & Sandygulova, A. (2021). A long-term study of robot-assisted therapy for children with severe autism and ADHD. Nazarbayev University Research Repository.
<https://research.nu.edu.kz/en/publications/a-long-term-study-of-robot-assisted-therapy-for-children-with-sev>