

Olga Polovynka, Danil Nikulin, Olga Dmitrieva

RESEARCH OF THE PERFORMANCE OF THE APRIORI GROUP ALGORITHMS ON BIG DATA

The using of associative rules search algorithms for working with unstructured data of large volumes are substantiated in the article. It is noted that the most popular associative rules search algorithms is the Apriori algorithm. The results of the analysis of the operation of linear algorithms for associative rules search with unstructured data of large volumes are presented. The most famous modifications of Apriori algorithms for finding associative rules are considered. Comparative characteristics of the considered algorithms are given, the strengths and weaknesses of each algorithm are indicated. The results of the research of the efficiency of linear algorithms AIS, Apriori and AprioriTID on the data of different volumes using computers with standard RAM volume are presented. It is found that when processing big data in the stated conditions, the operation of the Apriori and AprioriTID algorithms becomes impossible due to lack of RAM. To solve this problem, a further analysis of the structural features of apriori group algorithms is carried out. According to its results, the AIS algorithm was selected for processing Big data on computers with a standard amount of memory. This algorithm work slower, but does not require downloading transactions to the computer's RAM, and therefore does not depend on its size. Based on the results of using the AIS algorithm for different amounts of data, it was concluded that the proposed implementation does not require a large amount of RAM, which is an advantage, but with large amounts of data it is quite slow and the time for implementation is unacceptable for practice. To solve this problem, it is proposed to turn to methods for constructing parallel or distributed structures.

Keywords: performance, unstructured data, associative rules search algorithms, Apriori, AIS, AprioriTID, Big data.

*Стаття надійшла _____
Рецензент: д.т.н. Годлевський М.Д.*

УДК 004.04

Сирмаміїх І.В., Таранов І.Р., Романюта М.О.

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ ДОСЯГНЕНЬ СТУДЕНТІВ НА NODE.JS З ІНТЕГРАЦІЄЮ FIREBASE

Впровадження інформаційних технологій відбувається практично у всіх галузях, але найбільш гостро вирішення цього питання вимагає система освіти, зокрема організація управління на факультетах закладів вищої освіти.

Стаття присвячена розробці програмного продукту для автоматизованого контролю досягнень студентів зі статистикою, звітністю, зручним і швидким форматом взаємодії, який замінить паперовий ручний формат обробки документів. Мета при розв'язуванні задачі стояла - використовувати прогресивні технології та інструментарій. Серверна платформа сервісу - Node.js на мові TypeScript з сервісами Firebase від Google. Клієнтська частина створена з використанням Vue.js (frontend фреймворк для створення Single page application (SPA) - односторінкових додатків), Nuxt.js (фреймворк для Server-Side Rendering (SSR) - серверного рендерингу), Typescript (строго типізована мова програмування веб-додатків). Взаємодія з сервером виконується за допомогою Representational State Transfer API (REST API). Основна реалізація - у веб-додаток. Уся необхідна звітність по

факультетах, групах, конкретних студентах, курсах, роках, і рівню конкурсів генерується автоматично і видається в зручному для аналізу вигляді.

Продукт розроблено для деканатів Донецького державного університету управління, але може гнучко адаптуватися до індивідуальних потреб інших освітніх закладів усіх рівнів акредитації.

Автоматизація контролю досягнень студентів зі статистикою, звітністю, зручним і швидким форматом взаємодії, з використанням прогресивних та найбільш ефективних засобів для створення веб-додатків, суттєво вплине на конкурентні переваги навчального закладу на ринку освітніх послуг. А підвищення швидкості і якості роботи з контингентом студентів зможе забезпечити надійність і цілісність управління в цілому.

Ключові слова: облік досягнень, семестровий контроль, ранжування, програмний продукт, серверна частина, клієнтська частина Node.js, TypeScript, Firebase, Serverless, SPA, Vue.js, Nuxt.js, REST API, веб-додаток.

Постановка проблеми. У даний час в освіті помітно збільшується потік найрізноманітнішої інформації, тому застосування інформаційних технологій допоможе вдосконалити засоби і способи її накопичування, передачі, обробки та зберігання, застосувати її в професійній діяльності. На жаль, технології вкрай повільно впроваджуються в документообіг, в звітність, в спілкування між студентами і викладачами, а також в навчальний процес.

Система освіти в ДонДУУ передбачає підведення підсумків семестрового контролю, ранжування всіх студентів. Ці рейтингові списки є визначальними для призначення стипендій студентам університету. 90% рейтингового балу студента складають навчальні досягнення (успішність), а 10% – досягнення, що характеризують участь студента у науковій, науково-технічній діяльності, громадському житті та спортивній діяльності (згідно затвердженого Переліку нараховуються додаткові бали за участь і перемоги в конкурсах, олімпіадах, спортивних змаганнях та інших видах позанавчальної діяльності). Саме облік досягнень студентів - одна з існуючих проблем в документообігу університету. Обробка документів виконується повністю в ручному режимі: починаючи з того, що студенти повинні самостійно принести копію документа, закінчуючи тим, що відповідальний співробітник розраховує суму балів по кожному зі студентів у кінці семестру. Така система ненадійна і неефективна з наступних причин: цей процес займає величезну кількість робочого часу співробітника і особистий час студентів; людський фактор може позбавити студента заслужених балів (він може забути принести диплом, диплом може загубитися в університеті протягом семестру, перевіряючий прорахується та ін.).

Тому розробка програмного продукту для автоматизованого контролю досягнень студентів зі статистикою, звітністю, зручним і швидким форматом взаємодії, з використанням прогресивних та найбільш ефективних засобів для створення веб-додатків, який прийде на зміну паперовому ручному формату обробки документів є актуальною.

Аналіз останніх досліджень і публікацій. Хоча і було проведено чимало досліджень, але й досі не створено оптимальну систему для забезпечення документообігу у ВНЗ. Причинами того є велика кількість різних типів документів, що використовуються в освітній установі, та швидкий розвиток сучасних технологій, що призводить до втрати актуальності досліджень, які проводилися ще кілька років тому. Також швидкий розвиток інформаційних технологій приводить до появи ґрунтовно нових технологій та засобів роботи з даними, наприклад «хмарні» обчислення, і не можна лишатись осторонь цього.

Значний внесок в розвиток та поширення програмного забезпечення для вищих навчальних закладів України внесли багато науковців, зокрема Пономаренко В.С. [1], Задорожна Н.Т. [2], Мироненко Д.С. [3], Поліновський П.П. [4], Різун Н.О. [6], Федусенко

О.В.[7]. Над створенням, впровадженням та удосконаленням програмного забезпечення для вищих навчальних закладів України працює, наприклад, колектив розробників програмного забезпечення ПП "Політек-СОФТ" [5]. Розроблений ними пакет програм «Деканат» призначений для автоматизації планування та обліку навчального процесу. Існує достатньо систем обліку оцінок та прогресу учнів шкіл, серед найпопулярніших – Щоденник. Але більшість них сфокусовані на розробку автоматизованих програмних засобів для об'єктивного оцінювання при поточному та підсумковому контролі знань студентів та фахівців. Тому проект з обліку позанавчальної активності студентів буде першим проектом такого роду, аналогів якому ще немає.

Мета (завдання) дослідження. Метою статті є дослідження існуючого процесу збору досягнень студентів, його контролю та нарахування балів у навчальному процесі університетів, а також створення відповідного сервісу, використовуючи актуальні технології в області розробки веб-додатків та серверів для оптимізації документообігу у сфері освіти під технічною назвою «DiploKor». Це має бути першим кроком до інформатизації та міграції взаємодії зі студентами у цифровий простір.

Основний матеріал дослідження. Стрімкий розвиток технологій дозволяє на абсолютно новому рівні створювати якісні продукти будь-якої складності, які будуть задовольняти потреби замовника. Але більшість існуючих і сучасних веб-сервісів використовують морально застарілі технології на Frontend (jQuery) та backend (Vanila PHP). З моменту їх створення (через кілька десятків років), з'явилися нові і прогресивні технології для ефективної розробки якісних веб-додатків. Однак, з огляду на те, що більшість існуючих продуктів досі використовують застарілі технології, це створює попит на фахівців такого профілю. Це і є однією з причин, чому нові продукти і зараз розробляються без використання нових рішень. Більш того, ці технології продовжують вивчатися в університетах, а нові - навіть не заплановані програмою А ринок, зі свого боку, перенасичується щороку тисячами програмістів без досвіду, лише зі знаннями застарілих неперспективних технологій.

Для розробки програмного продукту для автоматизованого контролю досягнень студентів платформою для серверної частини було обрано Node.js сервер на мові програмування TypeScript з сервісами Firebase від Google, серед яких: Authentication, Firestore, Functions (serverless рішення для Node.js), Storage, Analytics, Hosting. В системі передбачено 3 категорії користувачів: *Студент, Модератор і Суддя*:

- Студент - при реєстрації вказує свої дані: ПІБ, факультет, групу і завантажує свою фотографію. Основний сценарій використання - відправка на перевірку фотографій своїх дипломів і грамот.
- Модератор - акаунт створюється після перевірки прав доступу особистості до даних студентів. Основний сценарій використання - перевірка нових студентів в системі з даними в базі студентів університету, щоб не допустити недійсні акаунти в системі або акаунти з невірними даними.
- Суддя - акаунт створюється після перевірки прав доступу особистості до даних студентів. Основний сценарій використання - звірка даних, надісланих студентами, з фотографією диплому.

Уся необхідна звітність по факультетах, групах, конкретних студентах, курсах, роках, і рівню конкурсів генерується автоматично і видається в зручному для аналізу вигляді.

Основний сценарій використання: студент завантажує фотографію з документом, вказує дату вручення, один з рівнів масштабу заходу (міський, Всеукраїнський, Міжнародний і т.д.), зайняте місце і є чи захід профільним. Суддя отримує сповіщення про те, що нові дипломи чекають на верифікацію. Він звіряє інформацію, зазначену студентом і на дипломі, і підтверджує, редагує і підтверджує, або відхиляє із зазначенням причини. В

кінці семестру Суддя отримує звітність по семестру: кількість додаткових балів для кожного зі студентів; кількість балів, отриманих за профілем студента; статистику по факультетах, групах і курсах для оцінки ефективності позанавчальної роботи. Аналогічну статистику можна отримати за допомогою одного з попередніх періодів, по яких наявна інформація. Студент може отримати доступ до всіх своїх дипломів за весь час навчання, перевіряти їх статус і бачити порівняльну статистику по іншим студентам.

При розробці програмного продукту для автоматизованого контролю досягнень студентів була поставлена задача використовувати прогресивні технології та інструментарії. Кожні мають свої позитивні і негативні сторони, тому після їх порівняльної характеристики, аналізу завдання, яке було поставлено університетом, і які KPI (Key Performance Indicator) повинні бути досягнуті, були відібрані наступні технології / фреймворки:

1. Серверною частиною програми є набір сервісів від Firebase:

- Authentication - надає готове рішення реалізації протоколу авторизації OAuth 2.0 з підтримкою наступних способів аутентифікації: Email & Password, phone number, Google, Play Games (Android), Game Center (iOS), Facebook, Twitter, GitHub, Yahoo, Microsoft, Anonymous.

У нашому продукті використовується аутентифікація за номером телефону, як найбільш оптимальний варіант, який зможе надати адміністрації університету простий і найшвидший спосіб зв'язку зі студентами. Також, наявність номера телефону найбільш ймовірна у кожного зі студентів, на відміну від інших сервісів аутентифікації.

- Storage - хмарне файлове сховище.

У нашому сервісі, в ньому зберігаються фотографії студентів і фотографії дипломів;

- Firestore - документно-орієнтована база даних.

Структура являє собою послідовне чергування колекцій і документів необмеженої глибини укладення. Підколекції є незалежними від документа, в якому вони знаходяться. Кожен рівень ієрархії бази даних може являти собою або набір колекцій, або набір документів. У ній зберігаються всі дані користувачів додатка, логи і аналітика по звітних періодах.

- Hosting - хостинг сайту, на якому і знаходиться frontend-частина сервісу.
- Functions - є serverless рішенням для розгортання Node.js додатку.

На ньому розташована вся серверна частина з обробки запитів користувачів і даних.

- Analytics - надає сукупність мікросервісів для аналітики активності користувачів в додатку.

Є бесшовно інтегрованим рішенням з Google Analytics. У нашому випадку використовується для відстеження ключових показників активності в додатку студентів.

2. Клієнтською частиною програми є набір засобів:

- Vue.js - прогресивний Frontend фреймворк, який призначений для реалізації користувацьких інтерфейсів.

Перш за все він вирішує завдання рівня уявлення. Vue.js, як і будь-який інший фреймворк, розроблений для звільнення від написання однотипного коду. Основні концепції це: конструктори, компоненти, директиви, переходи.

- Nuxt.js - це фреймворк для фреймворка, призначений для створення універсальних додатків на Vue.js, де Backend частиною є Node.js.

Головна ідея цього фреймворка полягає в рендеринге UI на стороні сервера, що позитивно впливає на індексування в пошукових системах. Також Nuxt.js з коробки пропонує зручні модулі для роутінга і управління станом додатку.

- TypeScript – статично типізована мова програмування, яка компілюється в JavaScript.

Так як фреймворк Vue.js використовує JavaScript, який є динамічно типізована мова програмування, то було прийняте рішення використовувати модуль, який дозволяє деякі частини програми (стан додатку, компоненти) написати з використанням TypeScript, щоб отримати строгу типізацію і підвищити читабельність коду.

- REST API – Representational State Transfer Application programming interface для обміну даними між Frontend і Backend частинами сервісу.

Вірніше буде використовуватися обгортка від Firebase сервісів. А саме Functions і Authentication, які реалізують принципи REST API.

- SPA - Single Page Application.

Односторінковий додаток, який працює повністю без перезавантаження сторінки. Всі дані, які були отримані з Backend зберігаються в Store (зберігає стан всього додатку). Перевагою цього принципу є те, що уникається завантаження одного і того ж контенту сторінки при переходах по сайту. Наприклад, footer, навігаційна панель, header і інші частини сторінки найчастіше ніколи не змінюються на сторінках сайту. Змінам при переходах схильний тільки основний контент сайту, при використанні SPA тільки він і доставляється на клієнт, наслідком є зниження навантаження на мережу в порівнянні з класичними сайтами, де при кожному переході довантажуються вся сторінка цілком.

- SSR - Server-Side Rendering.

При використанні SPA на клієнт надходить тільки JavaScript код, який браузер сам повинен виконати і отримати HTML-розмітку. Такий спосіб працює досить швидко, але має недоліки у вигляді неможливості індексування контенту пошуковими роботами, тому що HTML-розмітка генерується тільки після отримання сторінки з JavaScript-ом і подальшими асинхронними викликами. SSR вирішує цю проблему, тому що вся HTML-розмітка генерується на сервері і віддається на клієнт відразу з контентом. Головним недоліком SSR є підвищене навантаження на сервер.

Найголовніше у цій системі – це її потенціал. Залучення найактивнішої частини студентів (які приймають участь у позанавчальних змаганнях) до цифрового виду взаємодії з університетом - це перший крок на шляху виведення системи освіти на новий, цифровий рівень. На основі електронних акаунтів студентів можна розширювати існуючу систему задля покращення навчального та позанавчального процесу. Також, цей продукт являє собою найшвидший спосіб інформування студентів, оскільки адміністрація буде мати можливість надсилати інформацію на телефон студента. Це значно швидше, ніж усі існуючі системи інформування в університетах: паперці на дошках, або передачі через старост (які також були запроваджені не менш ніж півстоліття тому). Тобто, якщо цей проект не буде мати перешкод з боку університету, це може стати багатофункційною системою цифрового персонального кабінету студента.

Поточний стан проекту – активна розробка. Декілька з вже реалізованих систем наведені нижче (рис.1 - рис.2):

- аутентифікація та авторизація;
- створення акаунта студента з його персональною інформацією;
- додавання нових модераторів;
- верифікація нових акаунтів студентів Модераторами;
- завантаження дипломів від студентів з вказанням мета-інформації;
- перевірка статусів усіх відправлених дипломів студентами;
- перевірка дипломів Суддею;
- отримання повної інформації по студенту Суддею з його історією відправки дипломів;
- розрахунок додаткових балів кожному зі студентів за семестр.

Серед наступних систем до реалізації: звітність, статистика, глибокий аналіз показників з формуванням висновків по групах, семестрах та студентах. Однією зі значних систем, що планується реалізуватись – гейміфікація процесу. Вона дозволить студентам порівнювати свій прогрес з іншими студентами та завдяки продуманому підходу спеціалісту у цій області – мотивувати студентів на досягнення більших результатів.

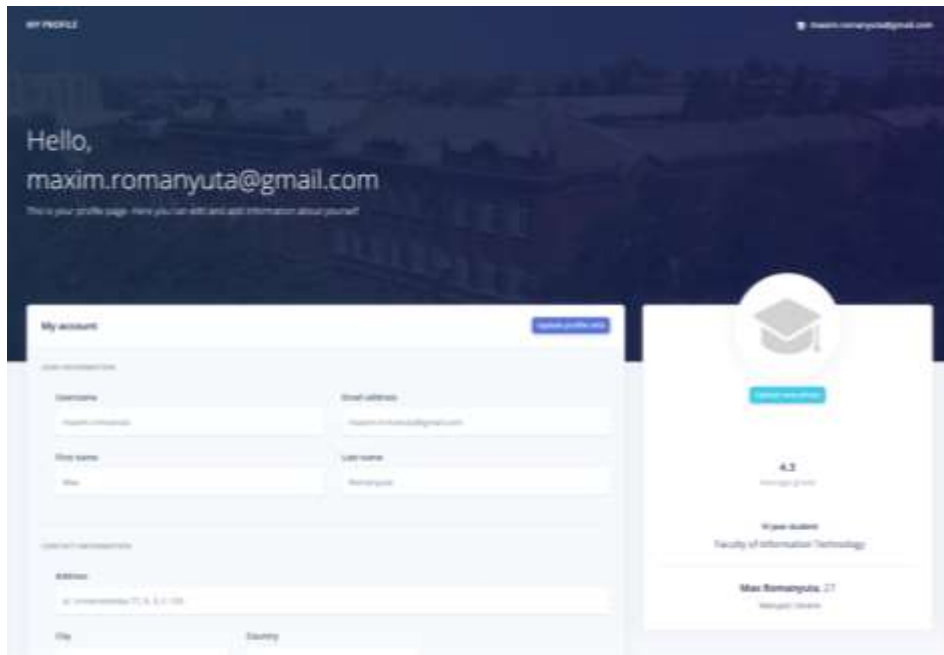


Рис 1. Редагування даних у особистому кабінеті Студента

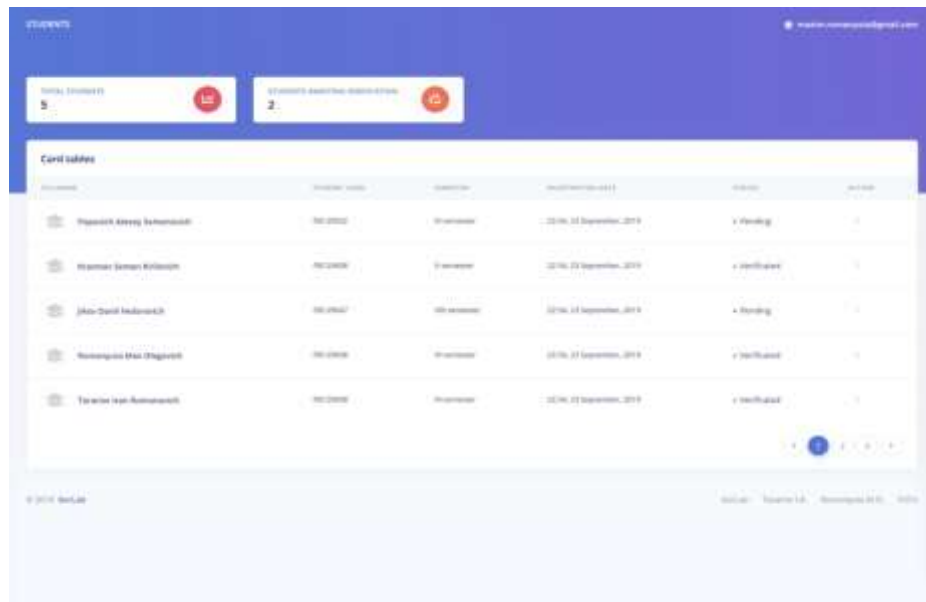


Рис.2. Список зареєстрованих студентів і їх статуси

ВИСНОВКИ

Таким чином, розроблена система допоможе зберегти особистий час студентів і робочий час співробітників, особливо в кінці семестру, коли співробітники повинні звітувати про

фактичне виконання свого індивідуального плану. Вона передбачає повну відсутність паперів в процесі оцінювання і зберігання досягнень студентів, мінімізацію ризику неврахування будь-якого з дипломів, а також підвищення зручності оцінки процесу. А головним критерієм у побудові було досягнення зручної і комфортної взаємодії між користувачем і технікою, враховувались ергономіка і пристрої, з якими буде взаємодіяти додаток (портативний комп'ютер, планшет або смартфон).

Список використаних джерел

1. Пат. 43616 Україна: МПК G09B 7/00. Автоматизована система тестування, навчання та моніторингу. / В.Д. Ціделко, Н.А. Яремчук, В.В. Шведова. Замовник та патентовласник: Національний технічний університет України "КПІ". – № 200902620; заявл. 23.03.2009; опубл. 25.08.2009.
2. Задорожна, Н. Т. Менеджмент документообігу в інформаційних системах освіти (для ВНЗ і ППО)/ Н. Т. Задорожна, К. М. Лаврішева. – Київ: Педагогічна думка, 2007.– 220 с.
3. Мироненко, Д.С. Автоматизированная система тестирования знаний студентов с многоуровневой архитектурой / Д.С. Мироненко// Системи обробки інформації: збірник наукових праць. – Х.: Харківський університет Повітряних Сил імені Івана Кожедуба, 2015. – Вип. 11 (136). – С. 215–220.
4. Поліновський, В. В. Впровадження системи електронного документообігу в науковій організації/ В. В. Поліновський, М. І. Огурцов // Вісник Хмельницького національного університету Технічні науки. – 2010. – № 4. – С. 117–123.
5. Політек-Софт. Програмне забезпечення для вищих навчальних закладів України [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: ПП "Політек-СОФТ", 1998-2019. – Режим доступу: www.politek-soft.kiev.ua(дата звернення 11.11.2019).
6. Різун, Н.О. Методика розробки автоматизованої системи управління якістю тестового контролю знань / Н.О. Різун, Ю.К. Тараненко // Вісник НТУ "ХПІ". Тематичний випуск: Інформатика і моделювання. – Х.: НТУ "ХПІ". – 2010. – № 31. – С. 145-152.
7. Федусенко, О. В. Системи управління навчальним процесом студентів з розгалуженою організацією дистанційного навчання у вищому навчальному закладі/ О. В. Федусенко, О. О. Рафальська// Київський національний університет будівництва і архітектури. – 2013. – № 13. – С. 162–165.

Сырмамиих И.В., Таранов И.Р., Романюта М.О.

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО УЧЕТА ДОСТИЖЕНИЙ СТУДЕНТОВ НА NODE.JS ПО ИНТЕГРАЦИИ FIREBASE

Внедрение информационных технологий происходит практически во всех отраслях, но наиболее остро решение этого вопроса требует система образования, в частности организация управления на факультетах высших учебных заведений.

Статья посвящена разработке программного продукта для автоматизированного контроля знаний студентов со статистикой, отчетностью, удобным и быстрым форматом взаимодействия, который заменит бумажный ручной формат обработки документов. Цель при решении задачи стояла - использовать прогрессивные технологии и инструментариум. Серверная платформа сервиса – сервер Node.js на языке TypeScript с интеграцией с сервисами Firebase от Google. Клиентская часть создана с использованием Vue.js (frontend фреймворк для создания SPA – одностраничных приложений), Next.js

(фреймворк для SSR - серверного рендеринга страниц), Typescript (строго типизированный язык программирования веб-приложений). Взаимодействие с сервером выполняется с помощью REST API. Основная реализация проекта – веб-приложение.

Вся необходимая отчетность по факультетам, группам, конкретным студентам, курсам, годам, и уровню конкурсов генерируется автоматически и выдается в удобном для анализа виде.

Продукт разработан для деканатов Донецкого государственного университета управления, но может гибко адаптироваться к индивидуальным потребностям других учебных учреждений всех уровней аккредитации.

Автоматизация контроля знаний студентов со статистикой, отчетностью, удобным и быстрым форматом взаимодействия, с использованием прогрессивных и наиболее эффективных средств для создания веб-приложений, существенно повлияет на конкурентные преимущества учебного заведения на рынке образовательных услуг. А повышение скорости и качества работы с контингентом студентов сможет обеспечить надежность и целостность управления в целом.

Ключевые слова: *учёт достижений, семестровый контроль, ранжирование, программный продукт, серверная часть, клиентская часть, Node.js, TypeScript, Firebase, Serverless, SPA, Vue.js, Nuxt.js. REST API, веб-приложение.*

Syrmamiikh I., Taranov I., Romaniuta M.

AUTOMATED STUDENTS ACHIEVEMENTS SYSTEM DEVELOPMENT ON NODE.JS AND INTEGRATION WITH FIREBASE

The introduction of information technology occurs in almost all industries, but the solution to this issue in the education system and particularly in the organization of management at faculties of higher educational institutions is the most acute.

The article is devoted to developing a software product for automated control of students achievement with statistics, reporting, convenient and fast format of interaction, which will replace the manual paper format of document processing. Service's Server Platform is Node.js with TypeScript programming language in integration with Google Firebase services. The client side is created using Vue.js (frontend framework for creating SPA - one-page applications), Nuxt.js (framework for SSR - server rendering), Typescript (strictly typed web application programming language). The server interacts with the REST API. The main implementation is in a web application.

All the necessary reports on faculties, groups, specific students, courses, years, and the level of competitions are automatically generated and issued in a form convenient for analysis.

The product is designed for deans' offices of the Donetsk State University of Management, but can flexibly adapt to the individual needs of other educational institutions of all levels of accreditation.

Automation of students' knowledge control with statistics, reporting, a convenient and fast interaction format, using the most advanced and most effective tools for creating web applications, will significantly affect the competitive advantages of an educational institution in the educational services market. And increasing the speed and quality of work with the student body will be able to ensure the reliability and integrity of management in general.

Keywords: *achievements Control, software product, Node.js, TypeScript, Firebase, Serverless, SPA, Vue.js, Nuxt.js.*

Рекомендовано до публікації: к. техн. наук Тузенко О.О.

Стаття надійшла _____.