CSIS 4495 - 002 - Applied Research Project

Final Report

**NATIVESPARK INTEGRATION SOLUTIONS CORP.**

**A logo with green leaves

Description automatically generated**

Valeriia Nikitina (300374609)

Video Link: <https://youtu.be/i0InDv3HUi0>

Contents

[Introduction 4](#_Toc195132603)

[The origin of the idea 4](#_Toc195132604)

[Problem framing 4](#_Toc195132605)

[Competitive advantage among others 5](#_Toc195132606)

[Overview of the platform 6](#_Toc195132607)

[The architecture of the system 8](#_Toc195132608)

[The hardware and software configuration of the system 8](#_Toc195132609)

[Hardware Requirements 8](#_Toc195132610)

[Software Configuration 8](#_Toc195132611)

[Changes made to the original proposal 10](#_Toc195132612)

[Features 10](#_Toc195132613)

[Technology 10](#_Toc195132614)

[Reasons for change 11](#_Toc195132615)

[Project Planning and Timeline 13](#_Toc195132616)

[**Planning and Proposal** 13](#_Toc195132617)

[**System Design** 13](#_Toc195132618)

[Development Phase 13](#_Toc195132619)

[Testing Phase 14](#_Toc195132620)

[Final Submission and Presentation 14](#_Toc195132621)

[Gantt Chart 15](#_Toc195132622)

[Implemented Features 16](#_Toc195132623)

[User Registration for All Three User Types 16](#_Toc195132624)

[Step 1: Basic Registration Form 18](#_Toc195132625)

[Step 2: Role-Specific Registration 19](#_Toc195132626)

[Password Encryption 22](#_Toc195132627)

[Existing Database Schema 22](#_Toc195132628)

[Login Handling 23](#_Toc195132629)

[Session-Based Authentication 24](#_Toc195132630)

[Welcome page 25](#_Toc195132631)

[Dynamic Navbar 26](#_Toc195132632)

[Python-based recommendation of products 27](#_Toc195132633)

[Evaluation Technique 28](#_Toc195132634)

[Reflections 29](#_Toc195132635)

[Hours logs 30](#_Toc195132636)

[References 35](#_Toc195132637)

[Downloads links: 35](#_Toc195132638)

[Appendix A: Installation Guide 37](#_Toc195132639)

[1. Start XAMPP: 37](#_Toc195132640)

[2. Open the NativeSpark Project: 37](#_Toc195132641)

[3. Run the Application: 37](#_Toc195132642)

[4. Run the Python Script: 37](#_Toc195132643)

[5. Access the Application: 38](#_Toc195132644)

[Appendix B: User Guide 39](#_Toc195132645)

# Introduction

## The origin of the idea

Originally developed by Valeriia Nikitina and Valeriya Saltykova, NativeSpark Integration Solutions Corp. emerged from their unique expertise and shared vision. Although the platform was initially a collaborative effort, Valeriia Nikitina will now continue to develop the idea and platform independently.

This transition marks a new chapter for NativeSpark, as Valeriia takes the lead in refining the platform's features, expanding its capabilities, and advancing its mission. By leveraging her background in data analysis and her commitment to empowering Indigenous communities, Valeriia aims to ensure that NativeSpark evolves into a robust, scalable solution that meets the current needs of its users while adapting to future challenges.

The project will maintain its original vision of fostering Indigenous entrepreneurship, with a strong focus on enhancing user experience, integrating advanced technologies, and exploring growth opportunities in the United States market. Valeriia is dedicated to preserving the collaborative spirit of the initiative while elevating the platform to new heights.

## Problem framing

NativeSpark is dedicated to helping Indigenous communities overcome barriers to participation in industry. Many Indigenous entrepreneurs lack access to essential resources, mentorship, and training necessary for successful competition [1]. Additionally, Indigenous products often struggle to reach mainstream markets, and cultural and regulatory obstacles further complicate participation.

Indigenous entrepreneurs face several systemic barriers that impede their ability to engage effectively in industrial and commercial opportunities, including:

* Many Indigenous communities are situated in remote areas, making it challenging for entrepreneurs to access the funding, tools, and infrastructure needed to grow their businesses.
* Entrepreneurs frequently lack mentors who understand their unique challenges, as well as training programs tailored to their cultural and business contexts.
* Indigenous products and services often encounter difficulties navigating regulatory frameworks and facing cultural biases that prevent them from accessing mainstream markets.
* Ongoing economic exclusion has perpetuated cycles of poverty, limiting opportunities for Indigenous entrepreneurs to thrive.

These challenges are further exacerbated by the impact of social determinants of health, as highlighted in research by Richmond et al. (2017) in Social Science & Medicine. The study underscores that structural inequities, including limited access to education, employment, and economic opportunities, disproportionately affect Indigenous communities. These inequities diminish quality of life and hinder participation in economic systems, perpetuating cycles of poverty in marginalized communities [2, 3].

## Competitive advantage among others

Existing platforms like the Indigenous Industrial and Contracting Network (IMCN) and Indigenous Business Development Services (IBDS) have made significant strides in supporting Indigenous entrepreneurs. However, they lack advanced technological features that could greatly enhance their effectiveness.

IMCN offers resources and networking opportunities but does not utilize AI-driven tools to optimize connections between businesses and entrepreneurs. On the other hand, IBDS emphasizes business development services but does not make use of predictive modeling or natural language processing (NLP) to improve decision-making and communication [4, 5].

These limitations hinder these platforms from effectively meeting the complex needs of Indigenous entrepreneurs, such as identifying market opportunities, facilitating collaboration, and offering personalized recommendations. NativeSpark aims to bridge these gaps by integrating:

* AI-Powered Matchmaking: Automatically connecting businesses with the most relevant Indigenous entrepreneurs based on their skills, location, and project requirements.
* NLP Capabilities: Enabling smooth communication across diverse linguistic and cultural backgrounds.

By addressing these gaps, NativeSpark strives to create a more comprehensive and inclusive platform for Indigenous entrepreneurs.

# Overview of the platform

The platform will serve as a bridge connecting three distinct types of users—Indigenous Peoples, businesses, and regular consumers—while also integrating an administrative role for monitoring and oversight. Its core mission is to empower Indigenous communities by providing a space to showcase their craftsmanship, connect with opportunities, and facilitate transactions in an engaging and supportive environment.

Indigenous Peoples will be central to the platform, showcasing their unique skills and products through detailed profiles that include craftsmanship information and photos. They can sell items individually or in bulk and browse job postings and bulk order requests from businesses. A social networking feature will promote visibility through likes and comments.

Businesses can place large-scale orders or post projects, specifying requirements like quantity and materials, and will pay a commission on transactions. Communication with sellers for customizations is encouraged.

Regular consumers can purchase unique, handmade items, also paying a commission. They can communicate with sellers for details or customizations.

The administrator role is vital for platform integrity, overseeing user activity, moderating content, and resolving disputes to ensure a positive experience.

Social Networking for Indigenous Profiles:

Users will create portfolio-like profiles with photos, descriptions, and reviews, promoting community engagement and talent recognition.

Custom Requests:

Business users have the option to request specific quantities of products using a custom order form linked from the product details page.

Recommendation Engine:

A Python-based recommendation system suggests similar products by utilizing a cosine similarity algorithm implemented with scikit-learn.

Entrepreneur Insights:

Entrepreneurs can access information about buyers who have purchased their products, helping to foster trust and facilitate communication.

AI-Powered Matchmaking:

AI will connect businesses with Indigenous individuals based on skills, ratings, location, and availability, while consumers receive personalized product recommendations using historical data.

Integrated Chatbot with Escalation to Admins:

A chatbot will address common inquiries and guide users, escalating unresolved issues to admins for further assistance.

Admin Dashboard:

Administrators will monitor transactions, moderate content, and manage chatbot escalations to ensure a safe and effective platform.

# The architecture of the system

In the development of NativeSpark's platform, we will adopt the Model-View-Controller (MVC) architecture model coupled with MySQL integration. This architectural approach is widely recognized and utilized in web application development due to its several advantages in terms of organization, scalability, and maintainability.

The MVC architecture separates an application into three interconnected components: Model, View, and Controller. Each component has specific responsibilities and interacts with the others to ensure efficient functionality and seamless user experience.

# The hardware and software configuration of the system

## Hardware Requirements

* Processor: Intel Core i5 or higher processors are recommended for smooth performance during development tasks.
* Memory (RAM): A minimum of 8GB RAM is recommended to handle the resource intensive tasks of running servers, databases, and development tools simultaneously.
* Network Connectivity: Stable internet connectivity is essential for accessing external resources, libraries, and version control systems during development.

## Software Configuration

1) Development Environment:

* Integrated Development Environment (IDE): IntelliJ IDEA will be used as the primary IDE for Java development. It provides robust features for Java programming, Spring framework support, and seamless integration with version control systems.
* Version Control: Git will be utilized for version control management, allowing collaborative development, code review, and version tracking.
* Build and Dependency Management: Maven will be used for managing dependencies and building the project.

2) Programming Languages and Frameworks:

* Java: The backend logic and business rules of the application will be developed using Java programming language.
* Spring Framework: Specifically, the Spring Boot framework will be used for rapid application development, dependency injection, and MVC architecture implementation.
* Thymeleaf: Thymeleaf will serve as the templating engine for generating dynamic HTML content in the View layer.
* Bootstrap: Bootstrap will be used for front-end styling, responsiveness, and user interface design.
* Python: Python is utilized for data seeding, database interactions, and developing a product recommendation engine.
* SQLAlchemy: This Python library is used for Object-Relational Mapping (ORM) access to MySQL database tables and models.
* Faker: Faker is a Python library that generates mock data (such as users, products, transactions, etc.) for testing and development purposes.
* scikit-learn: This is Python’s machine learning library, which is employed to implement the product recommendation engine based on similarity metrics.
* JavaScript: JavaScript allows for interactive client-side features, such as save/unsave toggle buttons and dynamic updates to the user interface.
* HTML & CSS: These technologies are used in conjunction with Thymeleaf and Bootstrap to create the markup structure and apply custom styling.

3) Database Management System (DBMS):

* MySQL: The MySQL relational database management system will be used to store and manage application data. It provides a robust and scalable solution for structured data storage.

## Changes made to the original proposal

### Features

|  |  |
| --- | --- |
| **Original** | **Final** |
| AI-Powered Matchmaking | Not implemented |
| Integrated Chatbot | Not implemented |
| Admin Dashboard | Not implemented |
| Social networking features | Not implemented |
| Custom product requests | Added |
| Saved listings for users | Added |
| Search, filter functionality | Added |
| Product recommendation system | Added |
| Registration of 3 user types | Done |
| Business (add/edit/delete job/project) | Done |
| Indigenous Entrepreneur (add/edit/delete product) | Done |
| Full product lifecycle (add to cart different items, checkout) | Done |

### Technology

|  |  |
| --- | --- |
| **Original** | **Final** |
| AI libraries: OpenAI API | Not used |
| Chatbot integration (Bootstrap widgets) | Not used |
| Python, Python-based tools | Added |
| Spring Boot, Thymeleaf, Bootstrap | Used |
| Programming languages (Java, JavaScript) | Used |
| MySQL | Used |

### Reasons for change

1. **AI Matchmaking and Chatbot**

Reason: The complexity of implementing these features, combined with limited time, made it challenging. They required natural language processing and third-party integrations (such as OpenAI), which would have significantly expanded the project's scope.

Justification: The focus shifted to completing the core platform functionality and product flows first before exploring advanced AI features.

1. **Admin Dashboard**

Reason: Priority was given to user-facing features to ensure that at list one end-user functionality was fully developed.

Justification: Given the scope and timeline of the semester project, the development of admin functionality was deferred to a later phase after project submission.

1. **Addition of a Python-Based Recommendation**

Reason: To provide meaningful user personalization without depending on OpenAI.

Justification: A lightweight, local solution utilizing cosine similarity and category/price metrics was implemented to offer relevant product suggestions, which was more feasible within the available timeframe.

1. **New Simple Features (Search, Filters, Saved Listings)**

Reason: These features were identified during development as critical for enhancing user experience.

Justification: Based on typical e-commerce platforms and feedback from peers and instructors, these features were prioritized to make the platform more functional and realistic.

1. **Technology Usage Shift**

Reason: Some proposed tools were deemed unnecessary or overly complex for the final project scope.

Justification: Simpler and more manageable solutions were adopted, such as writing the recommendation engine in Python instead of using external APIs.

1. **Custom Product Order Requests Feature**

Reason: Businesses have expressed a need to request specific quantities or variations of products directly from Indigenous sellers. Additionally, when stock runs out, basic users and other Indigenous entrepreneurs should have the option to place orders for custom quantities.

Justification: This feature enhances support for bulk orders and B2B interactions, aligning with the platform's mission to empower Indigenous entrepreneurship by fostering direct and customizable engagement between businesses.

# Project Planning and Timeline

## **Planning and Proposal**

**Milestone:** Proposal Approval (Jan 26)

**Deliverables:** Approved proposal document, initial project plan, and timeline allocation.

## **System Design**

**Milestone:** Finalize System Architecture (Feb 9)

**Deliverables:** Database schema and Interface design.

## **Development Phase**

**Milestone:** Initial Working Demo (Feb 24)

**Deliverables:** Partially functional prototype demonstrating core features and database integration, Video showing a demo of the implementation, midterm report.

**Milestone:** Advanced Feature Implementation (Mar 30)

**Deliverables:**

1. **User Authentication and Role Management:**

* Registration and login for users;
* Role-based access control: Entrepreneur, Buyer, Business;
* Dynamic navbar and page visibility based on user role.

1. **Product Listing & Details:**

* Products shown on the home page;
* Each product links to a detail page.

1. **Saved Listings Feature:**

* Save/unsave functionality for:
  + Products;
  + Jobs;
  + Projects;
* Saved items visible in a "Saved" section per user.

1. **Job and Project Postings:**

* Jobs and Projects posted by businesses visible to Entrepreneurs.

1. **Search Functionality:**

* Keyword-based search filtering.

1. **Filtering:**

* Products filterable by dynamically loaded categories from the database;
* Jobs filterable by employment type (Full-time, Part-time, Internship).

1. **Custom Order Request Form:**

* Businesses and other users can submit requests for custom product quantities.

1. **Recommendation Of The Product;**
2. **Full Product Lifecycle: From Browsing to Purchase:**

* **Browsing Products:**
* **Adding to Cart:**
* **Cart Management:**
* **Checkout Process:**
* **Order History & Details.**

## Testing Phase

**Milestone:** Refinement, testing and design (April 6)

**Deliverables:** Usability testing and revision.

## **Final Submission and Presentation**

**Milestone:** Final Submission and Presentation (Apr 13)

**Deliverables:** Functional platform, final report.

# Gantt Chart

A white calendar with blue text

AI-generated content may be incorrect.

# Implemented Features

## User Registration for All Three User Types

The platform supports registration for three different user roles:

* **Basic User**
* **Business User**
* **Entrepreneur User**

The registration process consists of **two steps**:

* **Step 1:** The user provides basic details such as email and password.
* **Step 2:** Additional details are collected based on the user type.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

### Step 1: Basic Registration Form

Each user registers with a unified form, where they select their role and provide credentials.

Example of Business Registration:

A screenshot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

### Step 2: Role-Specific Registration

After completing Step 1, users fill in additional details.

* **Basic User** – Provides a name and profile picture.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

* **Business User** – Adds business name, description, and logo.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

* **Entrepreneur User** – Provides personal and professional details.

A screenshot of a registration form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Password Encryption

To enhance security, all passwords are hashed using BCrypt prior to being stored in the database. This method prevents the storage of plaintext passwords, thereby reducing the risk of data breaches.

A computer screen with colorful text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Existing Database Schema

The platform follows database normalization (3NF), ensuring that user data is efficiently stored without redundancy.

A screenshot of a computer

AI-generated content may be incorrect.

## Login Handling

A screenshot of a login screen

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

## Session-Based Authentication

* **If a user is logged in**, they can browse restricted pages.
* **If not logged in**, they are redirected to the login page.

A computer screen shot of a program code

AI-generated content may be incorrect.

## Welcome page

A screenshot of a website

AI-generated content may be incorrect.

A landscape with mountains and a lake

AI-generated content may be incorrect.

A close-up of a magnifying glass

AI-generated content may be incorrect.

## Dynamic Navbar

If the user is not logged in, the "My Account" option is hidden in the navigation bar.

## Python-based recommendation of products

# Evaluation Technique

# Reflections

# Hours logs

|  |  |  |
| --- | --- | --- |
| Date | # Hours | Description of work |
| 12.01.2025 | 1 | I spent time brainstorming the main idea for my project by reviewing my past work to analyze its strengths and identify areas for improvement. I focused on finding a completely new scope of work that could add unique value and elevate the project's impact. |
| 17.01.2025 | 2 | I wrote a project proposal and conducted research to establish the domain, define the problem. Created project contract. |
| 20.01.2025 | 3 | Explore existing literature to identify knowledge gaps. This included developing hypotheses, outlining the research design, methodology, technologies, and expected results, emphasizing how the findings could address practical challenges and contribute to the field. Created Repo on GitHub. |
| 21.01.2025 | 2 | Reading different articles, watching YouTube videos about regarding AI and chatbot implementation. |
| 23.01.2025 | 1,5 | I wrote a project proposal (section project planning and timeline). Creating Gantt Chart. |
| 03.02.2025 | 2 | I started by creating my Spring Boot project and setting up the pom.xml file, ensuring the correct dependencies were included for the initial setup. I added a welcome.html page as the main landing page and configured a HomeController to serve it properly. Along the way, I encountered SQL database connection issues, which I resolved by correctly configuring application.properties with the necessary MySQL settingt. I implemented SecurityConfig.java, allowing public access to specific pages so that I wouldn’t need to log in every time. Additionally, I had issues displaying images, which I resolved by placing them in the static/images/ directory and updating the image paths in the HTML. |
| 04.02.2025 | 1 | I worked on integrating my IntelliJ IDEA project with an existing GitHub repository through the desctop version, ensuring proper Git initialization, remote repository linking, and successfully committing and pushing my code. |
| 05.02.2025 | 1.5 | I worked on refining the user registration process for the login page in my application. After users click "Register" on the login page, they are redirected to a user type selection page. I configured Spring Security to allow access to the /select-user-type endpoint, ensuring that users can view this page without needing to authenticate. Additionally, I updated the SecurityConfig to manage redirections correctly and resolved an HTTP 403 (Forbidden) error by adjusting the necessary permissions. I also verified that the "Go Back" button on the selection page accurately redirects users back to the login page. |
| 07.02.2025 | 2.5 | I worked on setting up the user registration flow for the NativeSpark project. Currently, I am implementing a multi-step registration process where users first select their type: Business, Indigenous Entrepreneur, or Basic User.  For business users, I have designed a two-step process. In Step 1, users provide their email, password, and user type, which are stored in the users table. In Step 2, users enter their business details and upload a logo, which is saved in the business\_users table. I have set up the necessary entities (User and BusinessUser), as well as repositories, services, and controllers to manage data persistence and form submission. Additionally, I created Thymeleaf templates for the registration pages.  I also spent time debugging issues related to database mapping. Despite these efforts, saving BusinessUser data in the database is still not functioning correctly, and I will need to continue troubleshooting this issue in the next session. I also plan to implement a similar process for the other two user types. |
| 09.02.2025 | 3 | I focused on debugging the issue where user data wasn't saved in the database after Step 2 of registration. Initially, there were errors with the @JoinColumn reference in the BusinessUser entity. After fixing this, I encountered another problem: uploaded logo files were not being saved correctly, leading to an error. I updated the BusinessUserService to ensure the upload directory exists before saving files. The application was trying to save files in a non-existent directory within the Tomcat temporary workspace. I modified the code to create the directory if it didn't exist and changed the file-saving logic to use `Files.copy()`, ensuring proper storage of uploaded files. I implemented the Entrepreneur User Registration process, following the same structure as the Business User Registration. Similar to the Business Registration, the entrepreneur registration consists of two steps.  In Step 1, the user enters their email and password, which are stored in the users table with the user\_type set to "ENTREPRENEUR." In Step 2, the user provides personal details, including their first name, last name, a brief description about themselves, indigenous identity, and a profile photo. This information is saved in the entrepreneur\_users table and is linked to the corresponding user. Writing logs and progress report 1. |
| 11.02.2025 | 2.5 | I implemented the Basic User Registration process, following the same structure as the Entrepreneur User Registration. Similar to the Entrepreneur Registration, the basic registration consists of two steps.  In Step 1, the user enters their email and password, which are stored in the users table with the user\_type set to "BASIC." In Step 2, the user provides personal details, including their first name, last name, a brief description about themselves, and a profile photo. This information is saved in the basic\_users table and is linked to the corresponding user. This time without debugging issues.  Initially, passwords were stored in plain text, but I needed to implement encryption using a PasswordEncoder. I updated the UserService class to include password encoding before saving user data to the database. However, I encountered an issue because UserService implemented UserDetailsService but did not override the `loadUserByUsername(String username)` method. This omission caused an error during the authentication setup in SecurityConfig.java. To resolve this, I implemented the missing `loadUserByUsername` method in UserService, ensuring that it retrieves user data from the database and returns a UserDetails object. This change enabled authentication to function properly with Spring Security. Additionally, I updated the authentication manager in SecurityConfig to utilize UserService for user authentication and to encode passwords before validation. Some modifications in htmls files. |
| 12.02.2025 | 1 | Today, I added a confirmation message for successful registration. However, I encountered an issue where the data from Step 2 of the business registration process was not being saved to the database. After debugging, I discovered that the problem was caused by the `event.preventDefault();` call in the form submission, which was preventing the form from sending data to the backend. To resolve this issue, I removed the `onsubmit` event. Once I verified that the business registration flow was working correctly, I implemented the same functionality for the Entrepreneur and Basic User registration processes to ensure consistency across all user types. New page subscription, need to connect it to all users. |
| 13.02.2025 | 2.5 | Styling home page. I was trying to implement subscription as step 3 of registration, but half way throug decided to leave for future account settings. |
| 14.02.2025 | 2 | Setting up the loging for users. Need to start again. |
| 16.02.2025 | 3 | Trial number two to set up login for users, unfortunately broke whatever had before. |
| 23.02.2025 | 3 | I have started from scratch developing the login access to the account for users, the programm logges in via users table.  At this trial i also had some issues, my login wasn’t working due to a conflict between the Authentication class from Apache Tomcat and the one from Spring Security. This conflict caused the authentication check to fail, preventing the user session from being recognized. Consequently, after logging in, the system redirected me back to the login page instead of taking me to the account page. To resolve this issue, I removed the incorrect import (org.apache.tomcat.util.net.openssl.ciphers.Authentication), ensured I was using org.springframework.security.core.Authentication, and updated my AccountController to properly verify if a user was authenticated using Spring Security. |
| 23.02.2025 | 2 | Writing Midterm Report, How to Run The Program document, recording Video and etc. |

# References

1. Statistics Canada. (2023, July 18). Indigenous entrepreneurship in Canada. Retrieved from <https://www150.statcan.gc.ca/n1/daily-quotidien/230718/dq230718c-eng.htm>
2. Richmond, C. A. M., & Cook, C. (2017). Creating conditions for Canadian Aboriginal health equity: The promise of healthy public policy. Social Science & Medicine, 176, 93–103.
3. Indigenous Corporate Training Inc. (n.d.). 8 key issues for Indigenous peoples in Canada. Retrieved from <https://www.ictinc.ca/blog/8-key-issues-for-indigenous-peoples-in-canada>
4. **Indigenous Industrial and Contracting Network. (n.d.). Retrieved from**[**https://www.imcn.ca/**](https://www.imcn.ca/)
5. **Indigenous Business Development Services. (n.d.). Retrieved from**[**https://ibdssk.com/**](https://ibdssk.com/)
6. Python Software Foundation. (n.d.). *venv — Creation of virtual environments*. Python 3 documentation. Retrieved April 9, 2025, from <https://docs.python.org/3/library/venv.html>
7. W3Schools. (n.d.). *W3Schools Online Web Tutorials*. Retrieved April 9, 2025, from <https://www.w3schools.com/>
8. OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. <https://chat.openai.com/chat>
9. Grammarly. (n.d.). Grammarly. Retrieved April 9, 2025, from <https://www.grammarly.com/>
10. DeepSeek. (n.d.). *DeepSeek: Exploring Uncharted Territories*. Retrieved April 9, 2025, from <https://www.deepseek.com/>

## Downloads links:

1. Python Software Foundation. (n.d.). *Download Python*. Retrieved April 9, 2025, from <https://www.python.org/downloads/>
2. Oracle Corporation. (n.d.). *MySQL: The world's most popular open source database*. Retrieved April 9, 2025, from <https://www.mysql.com/>
3. JetBrains. (n.d.). *IntelliJ IDEA: The IDE for Pro Java and Kotlin Development*. Retrieved April 9, 2025, from <https://www.jetbrains.com/idea/>
4. Apache Friends. (n.d.). *Apache Friends - XAMPP*. Retrieved April 9, 2025, from <https://www.apachefriends.org/>

# Appendix A: Installation Guide

### Start XAMPP:

* + Open the XAMPP control panel.
  + Start the **Apache** and **MySQL** services.
  + A screenshot of a computer

    Description automatically generated

### Open the NativeSpark Project:

* + Launch **IntelliJ IDEA**.
  + Open the NativeSpark project

### Run the Application:

* + Click the **Run** button in IntelliJ IDEA to start the application.

A grey background with white text

AI-generated content may be incorrect.

* + In case of the warning like below, please press “Fix auto”

A screenshot of a computer error

AI-generated content may be incorrect.

### Run the Python Script:

* + Open the Terminal and run the following commands:
  + cd desktop cd W25\_4495\_S2\_ValeriiaN
  + cd reccomendations Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser
  + venv/Scripts/activate
  + pip install -r requirements
  + cd src
  + python basic.py
  + python -m db.seed (if needed to populate the database with fake data)
  + python basic.py (repeat the following command to update the product recommendations based on the added products to the database)

### Access the Application:

* + Open a web browser.
  + Navigate to http://localhost:8082 to access the NativeSpark application.



# Appendix B: User Guide