ALLAM®

Challenge

تحدي علّام

ARABOT

ارابوت

تحدي علّام 24

المرحلة النهائية

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy

Overview

Arabot is a comprehensive platform aimed at teaching the Arabic language to both adults and children, catering to beginners as well as advanced learners.

The website features an extensive library of videos and books specifically designed to teach children in an interactive and accessible manner. In addition to this, it includes advanced educational tools such as a unique tool for grammar parsing and essential language rules, which help learners understand the language deeply and simply.

Furthermore, the project integrates a dedicated section for teaching the Quran, allowing visitors to read the Quran and learn the correct recitation, along with hadith and dhikr (remembrance of Allah). This makes it a holistic tool for learning both the Arabic language and Islamic teachings.

ALLAM Challenge
تحدي عـــلّام

ARABOT
اراسبوت

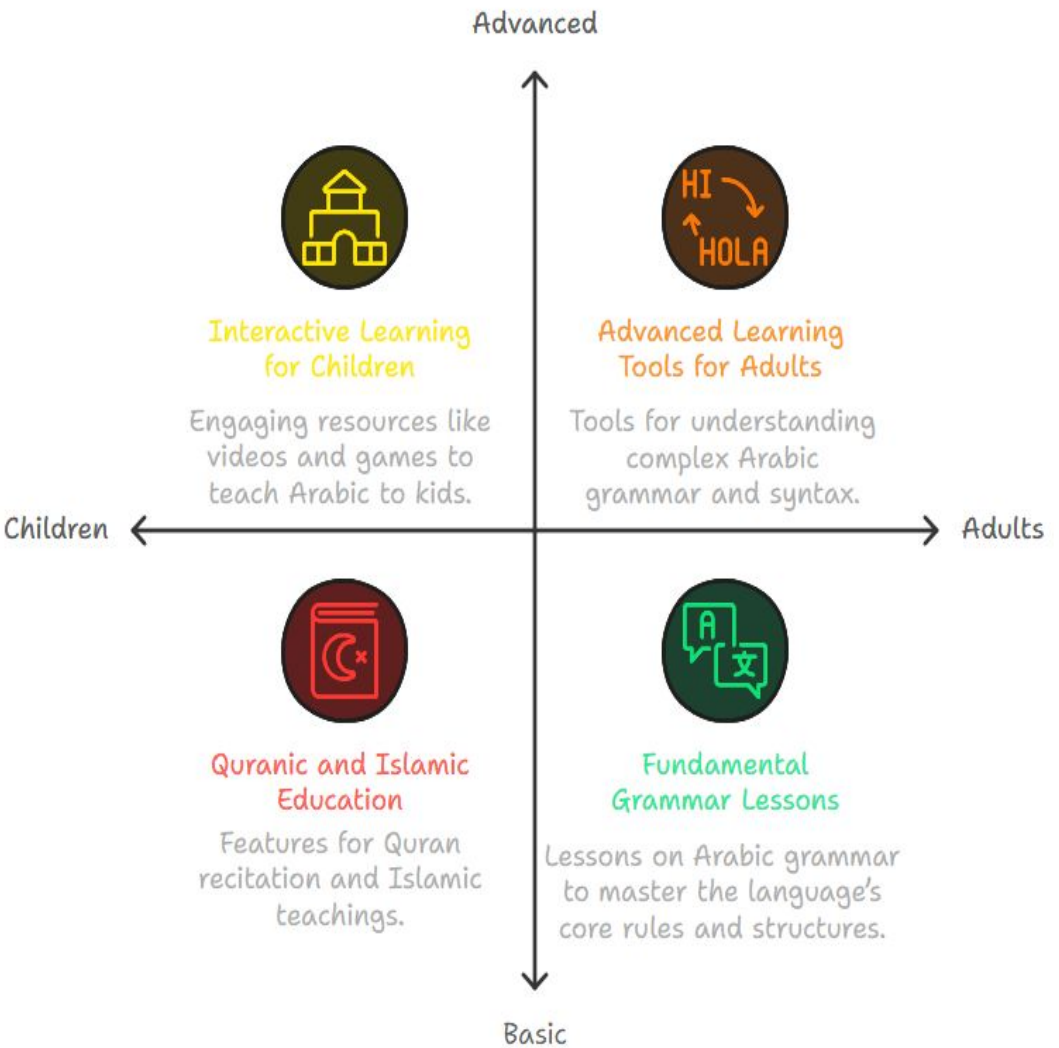
Table of Contents

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy

Features


Advanced
Basic
Children
Adults
HI
HOLA
Interactive Learning for Children
Engaging resources like videos and games to teach Arabic to kids.
Advanced Learning Tools for Adults
Tools for understanding complex Arabic grammar and syntax.
Quranic and Islamic Education
Features for Quran recitation and Islamic teachings.
Fundamental Grammar Lessons
Lessons on Arabic grammar to master the language's core rules and structures.

ALLAM Challenge
تحدي عـــلّام

ARABOT
اراسبوت

Technologies

- Figma is used for UI/UX design and prototyping.
- Angular is used for building a dynamic UI.
- ASP.NET Core and EF Core manage server-side logic and data.
- SQL Server stores and manages user and content data.
- NLP models and speech recognition enhance learning and feedback.
- Git/GitHub facilitates collaboration and code management.
- Quran.com API provides access to Quranic content.

ALLAM Challenge
تحدي عـــلّام

ARABOT
ارابوت

Methodology

1. Research and Planning

Literature Review: Conducted a thorough review of existing educational resources and methods for teaching Arabic to both children and adults.
Needs Assessment: Engaged with potential users (teachers, students, and parents) to understand their needs and preferences in learning Arabic.

2. Design and Development

Architecture Design: Defined the software architecture using a Clean Architecture approach, ensuring scalability and maintainability.
API Development: Develop RESTful APIs for the application, enabling efficient data handling and interaction.
Model Selection: Integrated various machine learning models, including text-to-speech and natural language processing tools, to enhance the interactivity of the platform.

3. Implementation

Frontend Development: Created a user-friendly interface for users to navigate educational resources seamlessly.
Backend Development: Implemented the backend services, ensuring robust data management and processing capabilities.
Integration: Integrated third-party services such as IBM Watson for natural language processing and machine learning models for speech synthesis.

4. Testing

User Testing: Engaged with a group of end-users to gather feedback on usability and effectiveness, making necessary adjustments based on their insights.

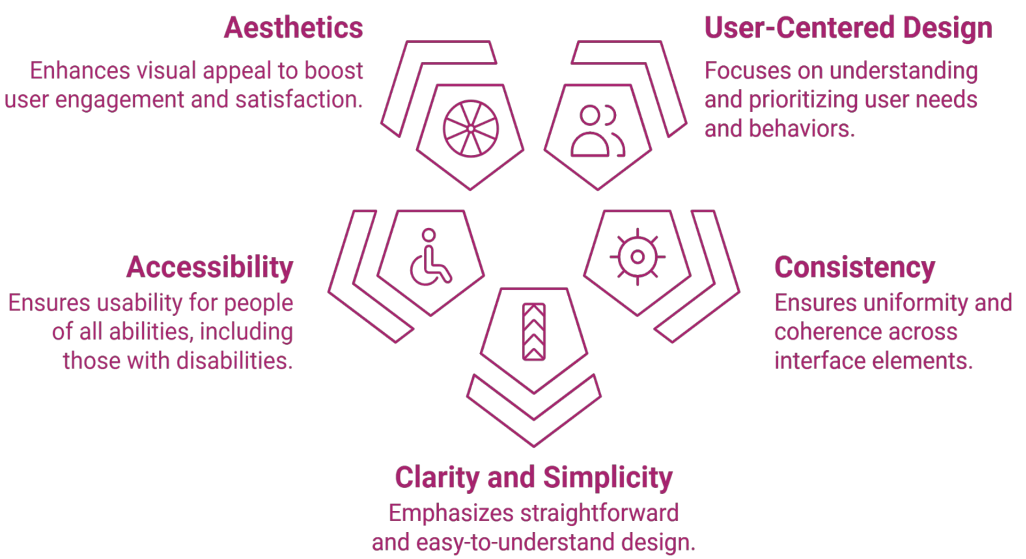
5. Deployment

Implementation

1. USER INTERFACE & USER EXPERIENCE

1. The Principles of UI/UX Design

User-Centered Design (UCD) focuses on meeting the needs, preferences, and behaviors of end-users throughout the design process, ensuring products are intuitive, efficient, and satisfying. Key principles of UCD include:



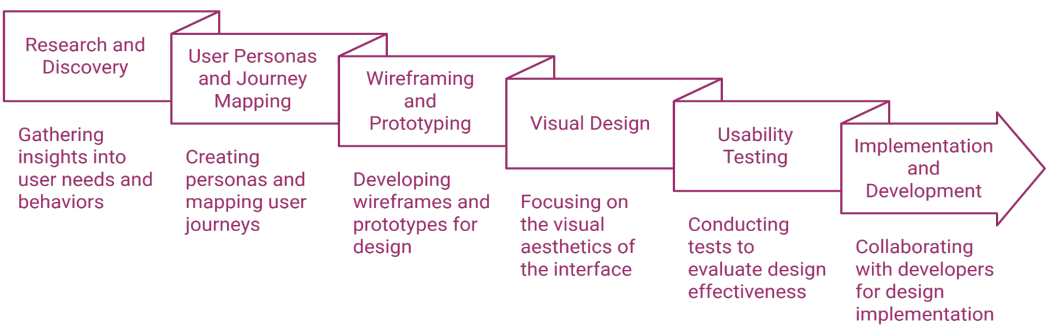
These principles help designers create products that are user-friendly, accessible, and visually appealing.

Implementation

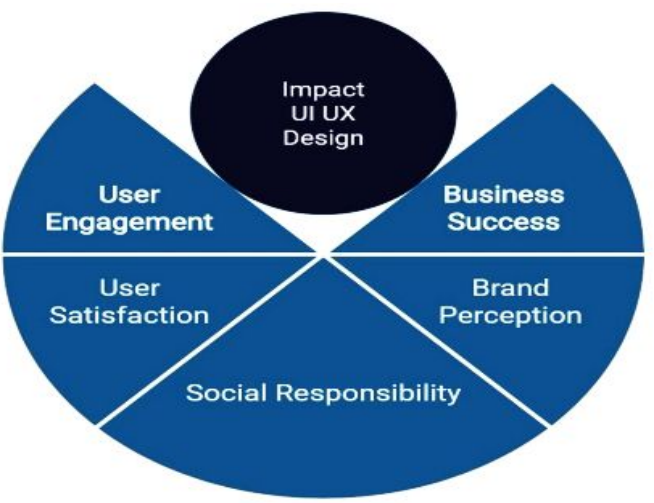
1. USER INTERFACE & USER EXPERIENCE

2. The Process of UI/UX Design

The process of UI and UX design involves a series of steps aimed at creating digital products and services that are intuitive, engaging, and user-friendly. While specific methodologies and frameworks may vary, the general process typically includes the following stages:



3. The Impact of UI/UX Design



ALLAM Challenge
تحدي عـــلّام

ARABOT ارابوت

Implementation

1. USER INTERFACE & USER EXPERIENCE

4. Process to create our UI/UX Design

Evaluate
Assess the product's usability and user satisfaction.

Implement
Build the design into a functional product.

Design
Develop detailed design solutions based on research and sketches.

Sketch
Generate initial ideas and concepts through low-fidelity sketches.

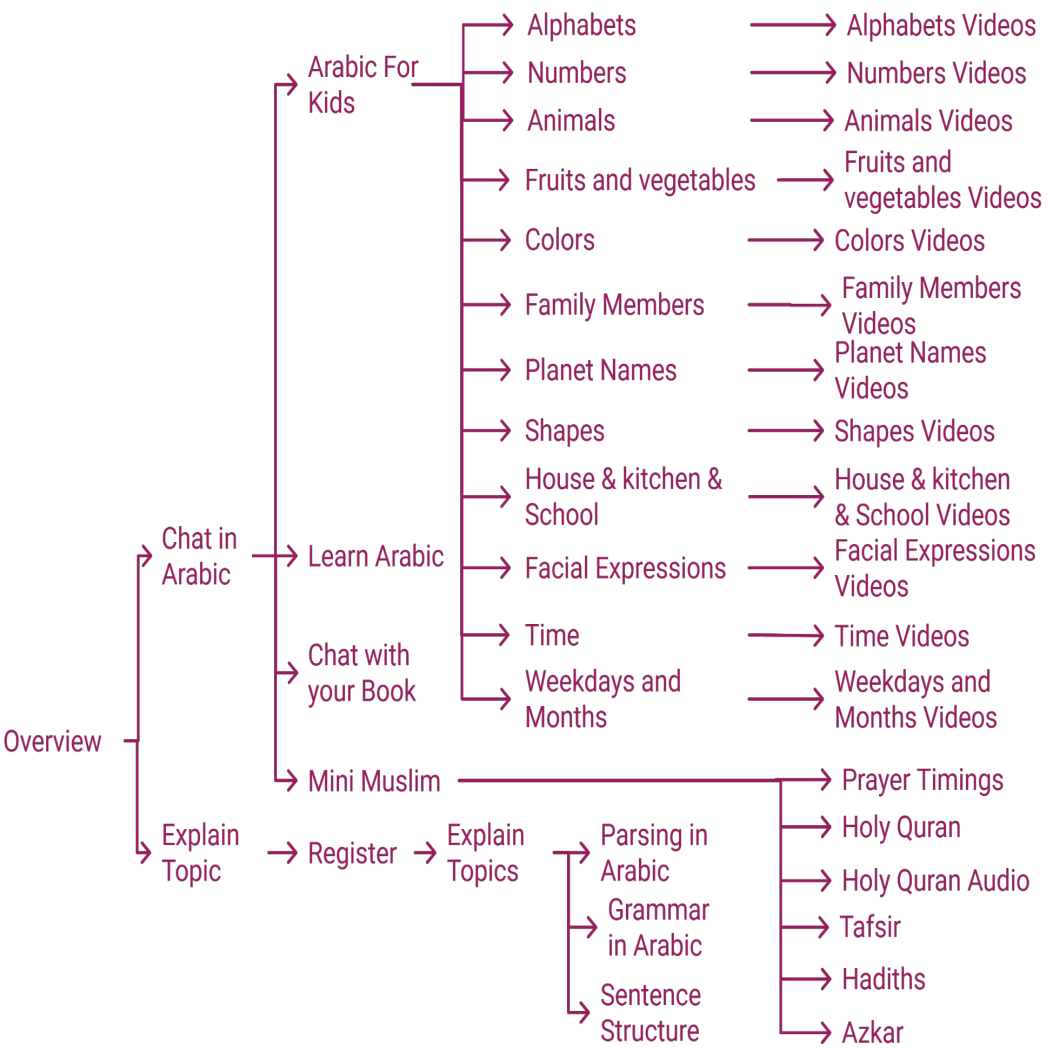
Research
Gather insights into user needs and preferences through various methods.

Understand
Gain a thorough understanding of project objectives and audience.

Implementation

1. USER INTERFACE & USER EXPERIENCE

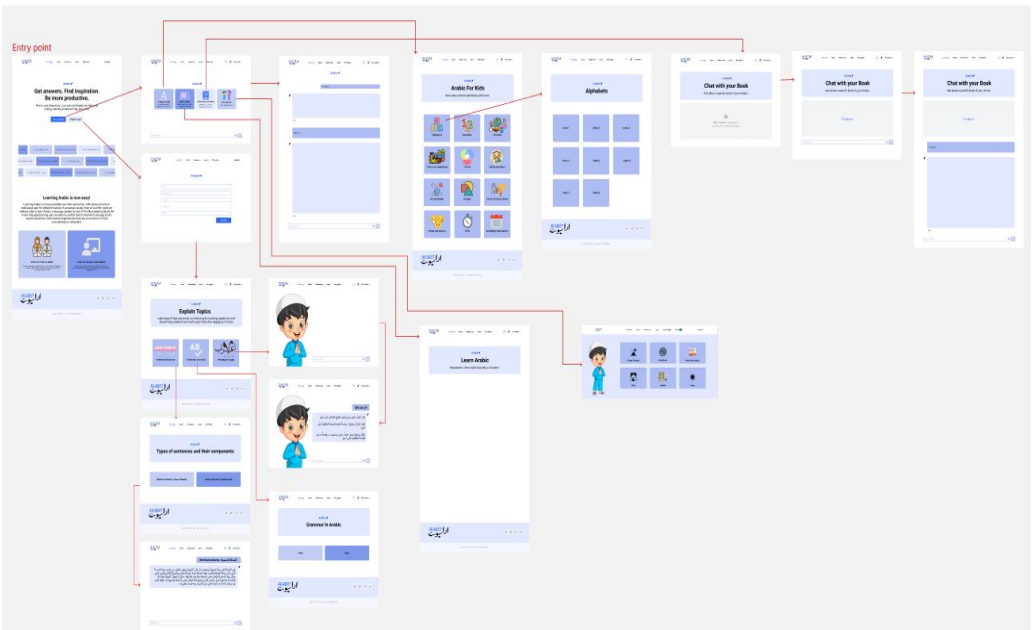
5. User Flow Diagram



Implementation

1. USER INTERFACE & USER EXPERIENCE

6. Screens of Web Application



After finalizing designs, they are handed off to developers for implementation. This phase involves translating design assets into code, integrating functionality, and building the final product. Designers collaborate with developers to ensure the design is accurately implemented and meets usability requirements.

Implementation refers to turning a design concept into a functional product or prototype using tools like coding, prototyping, or wireframing. Prototypes are simple versions of the final product used for testing and validation before sharing the final designs with developers. Prototypes will be created for each feature and presented for review.

Implementation

2. FRONT-END

Front-End Development

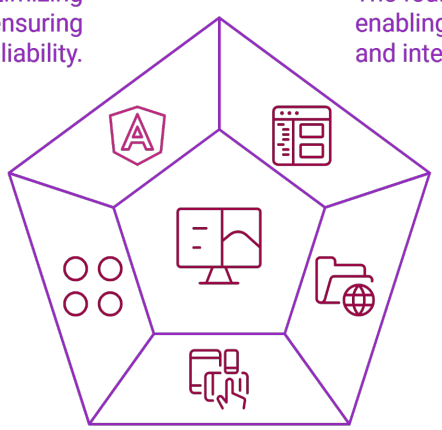
The front-end development of the Arabot application focuses on delivering an engaging user experience using core web technologies like HTML, CSS, and JavaScript, alongside Angular, a robust framework for building scalable web applications. Tailwind CSS is utilized for custom, utility-first styling, ensuring flexibility and responsiveness in design. Angular's services and RxJS handle state management, with additional support from NgRx for efficient data flow. Angular Router enables seamless navigation, and tools like Angular CLI and Webpack optimize the development workflow. To ensure quality, testing frameworks like Jasmine, Karma, and Protractor are integrated for comprehensive testing.

Build Tools & Testing

Tools optimizing workflows and ensuring application reliability.

Core Technologies

The foundational languages enabling structured content and interactivity.



State Management

Solutions like NgRx ensuring predictable data flow.

Frameworks & Libraries

Tools like Angular enhancing development efficiency and scalability.

CSS Frameworks

Tailwind CSS providing flexible, utility-first styling solutions.

Implementation

3. BACK-END

Securing the API with Identity Framework

In the process of securing the API with Identity Framework, the protection of Arabot's endpoints and resources against unauthorized access and malicious activities is enhanced. Identity Framework, part of ASP.NET Core, offers robust authentication and authorization mechanisms. Key aspects include:

- **Authentication**: It supports various schemes such as cookie-based authentication, JWT (JSON Web Tokens), OAuth, and OpenID Connect, requiring users to provide valid credentials or tokens to access protected endpoints.
- **Authorization**: After authentication, it enables fine-grained authorization based on user roles, claims, or contextual factors to control access to specific resources.
- **Identity Management**: Features for user identity management include registration, password management, account lockout, and profile management, enhancing security through password policies.

Implementation measures include configuring authentication middleware in the ASP.NET Core pipeline, defining authorization policies, applying attribute-based authorization, and ensuring secure transmission of credentials via HTTPS. The demonstration involves injecting the `UserManager<ApplicationUser>` and `IConfiguration` services to retrieve user information and manage JWT token generation. Upon successful authentication, claims are constructed, and a `JwtSecurityToken` is generated and returned in JSON format, establishing a secure API for Arabot and enabling clients to authenticate and authorize themselves using JWT tokens. The subsequent section will explore the process of documenting the API using Swagger.

Implementation

4. PROMPT ENGINEERING

Prompt engineering is central to AraBot's functionality, as it guides the IBM Watson model to produce structured, contextually relevant responses for Arabic grammar analysis. This involves designing a prompt that instructs the model to:

1. Recognize and understand Arabic sentences.
2. Identify grammatical elements.
3. Provide accurate parsing in a step-by-step format.

The current prompt aims to establish AraBot as a friendly and knowledgeable assistant for Arabic grammar (اعراب), leading the model to generate accurate grammatical analyses for each word in a given sentence.

1. IBM Watson Model Wrapper

The IBMWatsonXAIWrapper class acts as a bridge to IBM Watson's AI model, managing authentication, response generation, and handling errors.

Key Parameters and Attributes:

1. **model_id:** Specifies the unique model used for Arabic language parsing, pre-trained for high accuracy.
2. **parameters:** Includes values for max_new_tokens, temperature, and top_p to control response length, creativity, and relevance.
3. **prompt engineering:** The code embeds a specially crafted system prompt that guides AraBot's responses, helping to break down grammar in a structured, educational format.

2. Prompt Structure

The prompt used in this code consists of two main parts:

1. **System Prompt (sysPrompt):** Provides clear instructions for how the model should behave and the specific steps for parsing Arabic grammar.
2. **User Prompt (question):** Dynamic input from the user, which AraBot processes according to the instructions in the system prompt.

Implementation

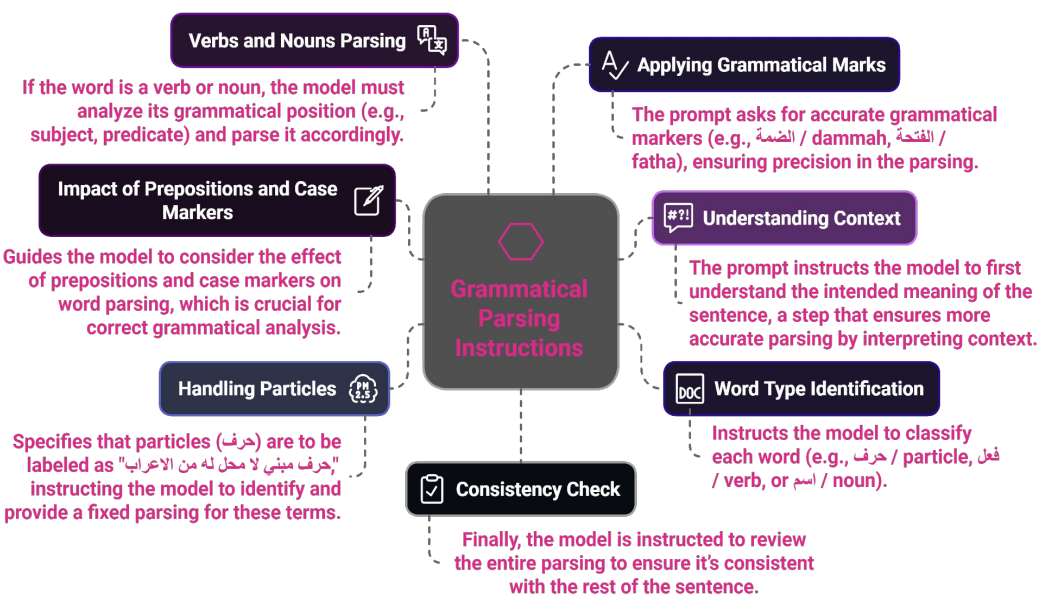
4. PROMPT ENGINEERING

3. Prompt Details

System Prompt (sysPrompt):
The system prompt is designed to establish AraBot's personality and its specialized role in Arabic grammar parsing. This section serves as the "persona" or identity of the model. Here's the breakdown:

sysPrompt = """<<SYS>> أنت "أرابوت"، مساعد الدردشة الآلي الودود والخبير في الاعراب في اللغة العربية. يتمثل دورك في مساعدة المستخدمين في اعراب جميع الكلمات والجمل التي يقدمها المستخدم. هذه المهمة تتطلب الخطوات التالية: ١- فهم المعنى المراد من الجملة ٢- تحديد نوع الكلمة (حرف، فعل، اسم) ٣- اذا كانت الكلمة حرف، اذا اعرابها "حرف مبني لا محل له من الاعراب" ٤- اذا كانت الكلمة فعل او اسم، قم بتحليل موقع الكلمة بالجملة ثم اعرب الكلمة حسب موقعها بالجملة ٥- إذا كانت الكلمة متأثرة بحرف جر أو أداة نصب او جزم، يجب تحديد تأثير ذلك على الإعراب ٦ - اعرب الكلمة بناءً على ما تم تحديده في الخطوات السابقة، وتأكد من استخدام العلامة الإعرابية المناسبة ٧- بعد إعراب الجملة، قم بمراجعة الإعراب للتأكد من اتساقه مع باقي الجملة <<SYS>> """

Explanation of Each Instruction Step:



Implementation

4. PROMPT ENGINEERING

This step-by-step structure helps the model provide responses that are not only detailed but also grammatically coherent.

User Prompt (question):
The question parameter is the user-provided text, a sentence that AraBot processes based on the structured instructions given in the system prompt. This prompt is prefixed with the system instructions and formatted in a specific way to ensure compatibility with IBM Watson's input expectations.

body = { "input": f"<s> [INST] {sysPrompt} {prompt} [/INST]", ... }

The combination of the sysPrompt and question forms the full prompt sent to IBM Watson, guiding it to produce a structured, grammar-parsed response.

4. Customization Tips

1. **Adjust Instructions for Different Grammar Levels:** The prompt could be adjusted to simplify grammar explanations for beginners or add complexity for advanced users.
2. **Add Explanations or Examples:** If the prompt needs to be more explanatory, add examples of parsing to the system prompt. For instance: ٨- إذا لم تكن الكلمة واضحة، قم بتوضيح القاعدة النحوية المتعلقة بالإعراب.
3. **Modify Language Style:** If needed for specific audiences, adjust the language style in the system prompt to be more formal or more conversational.
4. **Expand Scope:** The system prompt can be adjusted to include more detailed grammatical aspects, such as dual forms, plurals, or rare grammatical cases, if necessary.

Implementation

4. PROMPT ENGINEERING

5. Example Prompt and Response Flow

1. **User Input (User Prompt):**
   "اعرب جملة 'الطالب المجتهد يدرس بجد"
2. **Combined Prompt Sent to IBM Watson:**
   <<SYS>>
   "أنت "أرابوت ...
   ...
   <<SYS>> اعرب جملة 'الطالب المجتهد يدرس بجد'
3. **Expected Response:**
   الطالب: مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة، المجتهد: صفة مرفوعة، يدرس: فعل مضارع مرفوع، بجد: جار ومجرور...

This response reflects the prompt's structured guidelines, providing a detailed parsing of each word.

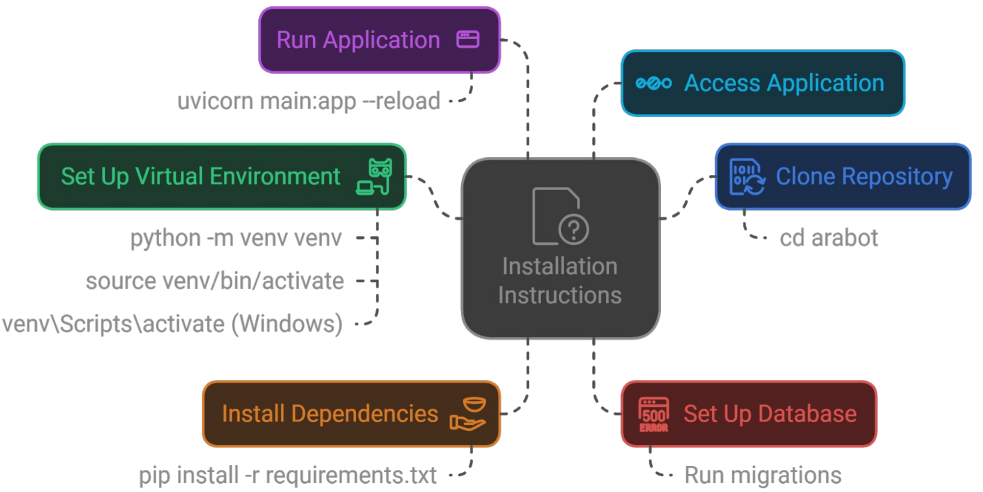
6. Sample Interaction on the Website

1. **User Enters:** "اعرب جملة 'الطالب المجتهد يدرس بجد'"
2. **AraBot Response:**
   الطالب: مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة
   المجتهد: صفة مرفوعة
   يدرس: فعل مضارع مرفوع
   بجد: جار ومجرور
3. **Educational Impact:**
   - **Improves Sentence Understanding**: Users see each word's role, reinforcing grammar rules.
   - **Encourages Self-Reflection:** By analyzing the feedback, users can apply grammar rules in future sentences.

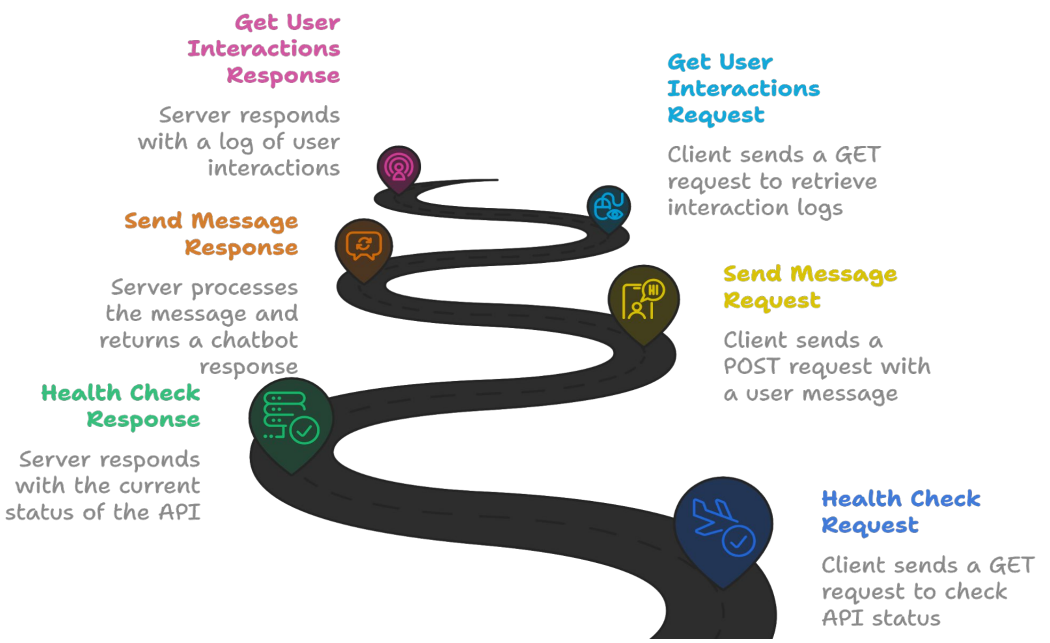
ALLAM Challenge
تحدي عـــلّام

ARABOT
اربوت

Installation



API Endpoints



الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

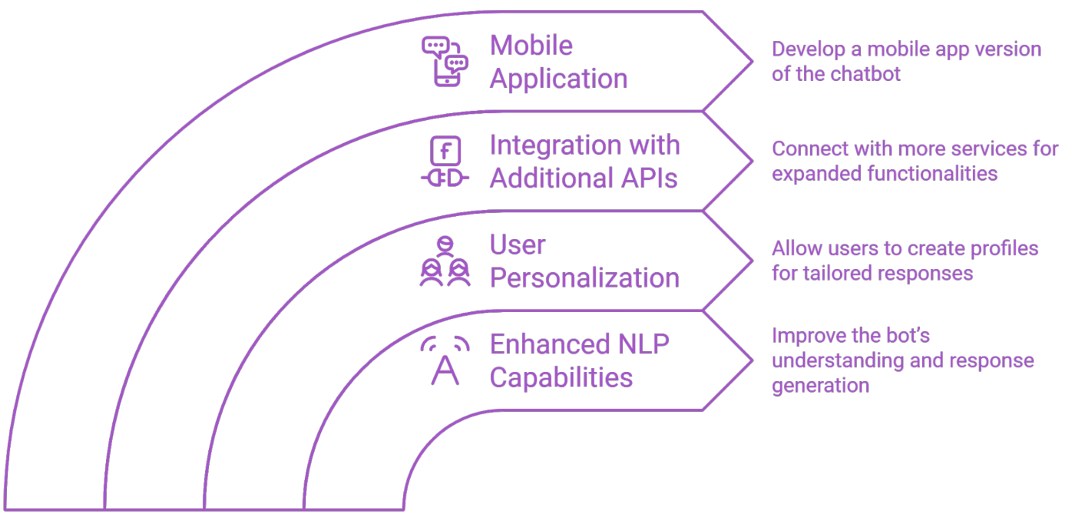
SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy

Future Enhancements

The system can be enhanced to:

- **Provide Grammar Tips:** Offer tips when users struggle with specific grammatical forms.
- **Include Interactive Grammar Exercises:** AraBot could quiz users on sentence components to strengthen retention.
- **Support Multiple Levels:** Adapt the system prompt to provide beginner, intermediate, or advanced parsing instructions.



ALLAM Challenge
تحدي عـــلّام

ARABOT
اربوت

Challenges

Throughout the development of the Arabot Project, several challenges were encountered:

1. Technical Challenges

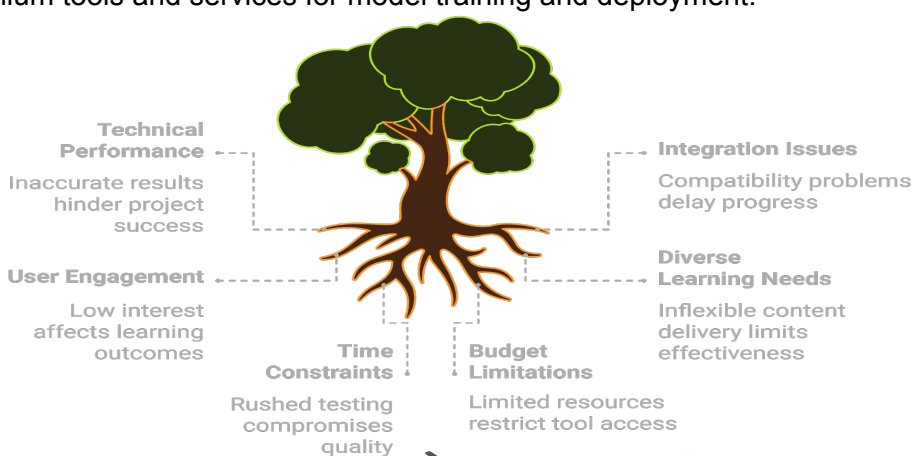
Model Performance: Ensuring the machine learning models delivered high-quality and accurate results, particularly in text-to-speech synthesis and natural language understanding.
Integration Issues: Integrating various APIs and services required extensive debugging and modifications to ensure compatibility and functionality.

2. User Experience Challenges

User Engagement: Designing an interactive platform that captivates users, especially children, posed a challenge in maintaining their interest and encouraging consistent learning.
Diverse Learning Needs: Addressing the varying levels of proficiency among users necessitated a flexible approach to content delivery and resource availability.

3. Resource Limitations

Time Constraints: Balancing the project timeline with thorough testing and user feedback led to pressure in meeting deadlines without compromising quality.
Budget Limitations: Constraints in funding limited the ability to access certain premium tools and services for model training and deployment.



ALLAM Challenge
تحدي عـــلّام

ARABOT أرابوت

Results

The Arabot Project has achieved several notable outcomes:

1. Functional Platform

Successfully developed a fully functional web application that provides a comprehensive learning resource for Arabic language learners, featuring video tutorials, interactive tools, and educational materials.

2. Positive User Feedback

Received encouraging feedback from users during testing phases, highlighting the platform's ease of use and effectiveness in teaching Arabic.

3. Enhanced Learning Experience

The integration of machine learning models improved the interactive nature of the platform, allowing for personalized learning experiences through features such as text-to-speech synthesis.

4. Contribution to Arabic Education

Positioned the Arabot Project as a valuable resource in the field of Arabic education, aiding learners of all ages in their language acquisition journey.

5. Future Potential

The project lays the groundwork for future enhancements, including the addition of new features, expansion of content, and continuous improvement of machine learning models based on user data and feedback.

ALLAM Challenge
تحدي عـــلّام

ARABOT ارابوت

Contributions and Acknowledgments

ALLAM Challenge – For developing such a professional competition where engineers can pursue their interests.

SDAIA – For sponsoring the challenge.

Islamic-API GitHub Repository – For providing essential resources related to Islamic teachings, enriching the educational content of the project, especially in Quranic studies and Islamic literature.

IBM – For providing a lot of resources and IBM watsonX.

Our mentors – For their continued technical support and encouragement.

Our families and friends – For their assistance in overcoming obstacles and achieving our objectives.

Discord and Google Documents – for assisting us in keeping track of our tasks and successfully functioning as a team.

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy

ALLAM Challenge
تحدي عـــلّام

ARABOT
اراسبوت

Thanks!

شـــكـــراً

الاتحـــاد الســـعـــودي للأمـــن السيـــبـــرانـي والبرمجة والدرونـــز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy