

Development of a Web-Based Expert System for the Diagnosis of Depressive Disorders

SVITLANA SOTNIK

Department of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering,
Kharkiv National University of Radioelectronics,
Kharkiv,
UKRAINE

Abstract: - The diagnosis of depression is a critical task in modern clinical psychology. However, traditional methods using paper questionnaires and manual scoring are labor-intensive and do not provide immediate feedback to patients. Existing web platforms for psychological testing are primarily focused on data collection and do not offer formalized algorithms for the automated interpretation of results according to clinically validated methods. This paper presents the development of a web-oriented expert system for diagnosing depression based on the Beck Depression Inventory. A mathematical model is proposed that formalizes the structure of the knowledge base as a set of questions with diagnostic coefficients, a user response vector, and a classification function for unambiguously mapping results to a set of diagnostic conclusions. Based on this formal model, a three-component system architecture was developed using React.js, which includes a root state management component, a component containing the expert system logic, and a static knowledge base module. The solution is implemented as a web application with an intuitive interface built using HTML5 and CSS3, along with data validation mechanisms and automated calculation of the diagnostic conclusion based on the clinically validated Beck Scale threshold values. Experimental testing on four response scenarios corresponding to different diagnostic categories confirmed the correctness of the algorithms for all intervals of the Beck Scale, including boundary values. The proposed approach provides immediate feedback to the user, increases the accessibility of screening diagnostics, and creates the prerequisites for timely seeking professional help through integration with the contact information of a specialist.

Key-Words: - Expert system, Depression diagnostics, Beck scale, Web application, Decision support, Mental health screening, Beck scale, Knowledge base, Automated diagnosis.

Received: April 19, 2026. Revised: May 11, 2026. Accepted: July 2, 2026. Published: October 23, 2026.

1 Introduction

1.1 Relevance and Purpose of Work

In the context of increasing psychosocial stress caused by social, economic, and personal factors, there is a steady trend towards a rise in the level of depressive disorders among the population [1], [2]. According to the World Health Organization, depression is one of the most common mental illnesses and significantly affects a person's quality of life, work performance, and social activity [3]. Timely detection of depression is critically important for effective treatment and the prevention of severe consequences.

In this study, the diagnosis of depression is understood as the process of identifying and assessing depression symptoms based on standardized psychometric methods, particularly the Beck Depression Inventory (BDI), which allows for

the quantitative determination of the severity of depressive symptoms.

The traditional diagnostic process is based on personal interviews by a clinician or the completion of paper questionnaires, followed by manual calculation of results [4]. In the absence of specialized decision support tools, this process is labor-intensive and requires significant time from the specialist. In addition, it does not provide immediate feedback to the patient, which reduces motivation to undergo diagnostics and complicates monitoring changes in the patient's condition over time.

Modern web technologies and software development approaches have proven effective for building information systems for a variety of purposes [5], [6]. This is particularly true for user interface libraries such as React.js, as well as for hypertext markup languages. Their flexibility and functionality enable the development of

accessible, user-friendly, and interactive tools designed to solve practical problems in various subject areas, including medical diagnostics. At the same time, existing psychology-oriented web applications are often not adapted to the specifics of clinical diagnostics, particularly to the use of validated questionnaires with clear interpretation algorithms. They also often fail to ensure adequate data confidentiality and user convenience.

This study combines aspects of computer science, software engineering, medical informatics, and clinical psychology, covering issues related to the development of web-based systems, user interface design, the implementation of data processing algorithms, and their application in psychometric diagnosis.

The relevance of this research stems from the need to develop a specialized web-based expert system designed to diagnose depression using a standardized methodology. Such a system must provide a user-friendly interface for taking the test, automated calculation of results according to predefined algorithms, immediate provision of interpreted results to the user, as well as storage of test history for further analysis of condition dynamics. Of particular importance is the integration of decision-making logic, based on clinically validated threshold values, into the system's software architecture.

The insufficient level of automation in mental health diagnostics leads to several issues. These include an increased workload for healthcare professionals, limited access to screening tools for the general public, difficulties in tracking changes in a patient's condition, and a higher likelihood of errors during manual processing. In addition, the lack of centralized web-based solutions with an intuitive interface complicates the conduct of mass screening studies and remote patient monitoring.

These challenges highlight the need to develop a software solution that combines modern web technologies to ensure accessibility and user-friendliness with a robust analytical framework to support diagnostic decision-making. The main obstacle is adapting general principles of web interface design to the specific requirements of medical diagnostics, which requires careful development of testing logic, response validation mechanisms, and correct interpretation of results in accordance with established clinical criteria.

Thus, there is a need to develop a web-based expert system for diagnosing depressive disorders that combines the functions of interactive questioning, automated result processing, and the provision of a diagnostic conclusion. Such a system

allows for structuring the process of collecting diagnostic information, reducing the influence of subjective factors in the interpretation of results, providing instant feedback to the user, and increasing the overall effectiveness of screening studies in the field of mental health.

This work aims to develop a web-based expert system designed to diagnose depression using the Beck Scale, which automates the testing process and increases the accessibility of screening diagnostics for users.

The main tasks are:

1. Analysis of existing methods and software tools for diagnosing depression, identifying their features and limitations.
2. Formalization of requirements for a web-based expert system for diagnosing depression.
3. Development of the architecture and operational algorithms for an expert system based on the Beck Scale.
4. Implementation of the software component of a web-based expert system, including the user interface and result processing logic.

1.2 Work-related Analysis

Modern web-based systems for psychological assessment are widely studied as tools for data collection, survey automation, and analytics [7], [8]. Their architectural features – the use of web interfaces, databases, modular structure, and scalability – create the conditions for the application of web technologies in medical informatics and clinical psychology. However, existing studies are mainly focused on the psychometric validation of online versions of questionnaires and do not consider the possibility of building open, software-integrated expert systems with formalized decision-making logic at the architectural level.

Traditional web-based survey systems, designed for data collection, are intended for the online presentation of questionnaires and the storage of responses in databases [9], [10], [11]. Their architecture is limited to sequential data collection, without formal integration of decision-making logic. In contrast, the proposed system incorporates a mathematically defined classification function directly into the processing pipeline, enabling immediate transformation of the response vector into a diagnostic conclusion. Such expert diagnostic tasks require not just the collection of responses, but their immediate automated processing according to clinically validated algorithms, where classification, result interpretation, and integration with a feedback

subsystem for further recommendations play a key role.

Despite functional differences, the basic web architecture provides a valuable foundation for specialized expert systems.

Existing platforms for psychological testing (Psytests [12], Mental Health Online [13], Qualtrics [14], Rozumka [15], Mamook [16] and similar systems) primarily focus on data collection and provide scoring through configurable modules. However, such implementations typically operate as closed or semi-transparent systems, where the decision-making logic is not explicitly formalized. In contrast, the proposed approach defines the diagnostic process as a deterministic mapping $f: A \rightarrow D$, embedded directly into the application code, ensuring transparency and reproducibility of results.

The authors in [17] describe Psytests as a platform for conducting psychological research and testing. Such systems offer advanced functionality for creating questionnaires, automating the collection of responses, and generating basic statistics.

The Qualtrics platform [18], [19] provides a powerful set of tools, including «Scoring» modules for assigning scores to responses, configuring interpretation ranges, and automatically generating results via «End of Survey Messages» or «Email Triggers». However, the implementation of interpretation logic in such systems is carried out through add-ons (scoring modules), which are part of a closed platform operating on a «black box» principle. This limits the ability to deeply modify the internal decision-making logic at the source code level and verify algorithm transparency, and also significantly complicates direct integration with other information systems.

The Mental Health Online platform [20] is a clinically oriented solution designed specifically to provide users with feedback on their test results. However, it operates as a closed ecosystem that does not allow for integration with specialists' local information systems or specific knowledge bases outside the platform's standard protocols. This makes it difficult for individual psychologists or small clinics to use, as they require close integration of diagnostic tools with their own websites and contact information.

The local platforms Rozumka [15] and Mamook [16] offer user-friendly builders for creating tests without the need for programming, making them accessible to a wide range of users. Their primary focus is on the ease of creating questionnaires and basic response statistics. At the same time, they do

not provide programming interfaces for implementing deterministic classification functions directly within the specialist's website architecture, which limits the ability to create a unified information space encompassing «diagnosis – contact – appointment booking».

An alternative to universal platforms is specialized clinical information systems, such as electronic Clinical Outcome Assessment (eCOA) platforms [21], including solutions from Medidata [22] and CRF Health [23]. A study [21] shows that eCOA systems are effective for data collection in clinical trials, ensuring compliance with regulatory requirements and high data quality. However, these systems were designed for different purposes and users. They are geared toward researchers and clinical trial sponsors. They do not include functionality to provide immediate feedback to the patient or recommend consulting a specialist with integrated contact information.

The studies reviewed in the field of online platforms for psychological testing demonstrate a wide variety of approaches: from universal survey builders to specialized clinical systems. However, these systems primarily focus on data collection and questionnaire management and lack formal mathematical grounding for diagnostic decision-making. In particular, they do not provide deterministic and reproducible inference mechanisms at the architectural level, which limits their applicability as true expert systems.

Unlike universal platforms (Qualtrics, Psytests), where the interpretation logic is implemented through add-ons (scoring modules), the proposed solution introduces a deterministic classification function integrated directly into the system architecture. While clinical platforms (Mental Health Online, eCOA) are closed ecosystems, the developed system is software-transparent, easily adaptable to specific clinical algorithms, and supports integration with specialists' local information systems.

The key advantage of the proposed approach is the formalization of the diagnostic process as a structured expert system, which enables both data collection and automated decision-making within a unified software architecture, ensuring a direct link between diagnosis and user guidance.

Novelty and Contribution Compared to Previous Work. Although certain background concepts (e.g., web application design and interface technologies) overlap with the author's previous conference publications, the present work introduces a substantially extended contribution. It includes a formal mathematical model of the diagnostic

process based on the Beck Scale, a deterministic classification function integrated into the system architecture, and a fully implemented three-component expert system with validated decision-making logic. Importantly, no previously published text, algorithms, or implementation details are reused in this manuscript. Compared to previous works focused on general web technologies and conceptual system design, this study provides a formalized, algorithmically defined expert system with experimentally validated behavior.

2 Formalization of Requirements for the Web-Oriented Expert System for Depression Diagnosis

The subject area of this study covers the process of automated psychological diagnosis of depression using web technologies. The system is based on the Beck Depression Inventory – a standardized psychometric instrument that allows for the quantitative assessment of the severity of depressive symptoms based on 21 questions, each of which corresponds to a specific clinical manifestation of depression. The term «question» refers to a separate group of statements from the Beck scale that corresponds to a single symptom of depression and contains four response options with different diagnostic coefficients. The official BDI methodology uses the term «group of statements», but in the context of the system's software implementation, both terms are used interchangeably, since each group is presented as a separate question with answer options.

Formally, the structure of the system's knowledge base can be described as a set of questions:

$$Q = \{q_1, q_2, \dots, q_j\}, \quad (1)$$

where each question q_i is an ordered pair:

$$q_i = \{t_i, C_i\}, \quad i = 1 \dots 21, \quad (2)$$

where i is the question number in the set Q ;

t_i is the text title of the question;

C_i is the set of answer options:

$$C_i = \{c_{i,0}, c_{i,1}, c_{i,2}, c_{i,3}\}, \quad (3)$$

where $j \in \{0, 1, 2, 3\}$ is the index of the answer option in the set C_i . This index is defined to be equal to its diagnostic coefficient, reflecting the severity of the corresponding symptom (0 – symptom absent, 3 – symptom maximally severe).

The set Q is implemented in the program code as an array of objects *data.quests*, where the *title* field corresponds to t_i , and the *choices* array corresponds to the set C_i . The last element of the array contains a completion marker with *title*: 'end' and is not a diagnostic question.

In the testing process, for each question q_i , the user selects exactly one answer option, which corresponds to a coefficient $a_i \in \{0, 1, 2, 3\}$. The numeric value of the coefficient is extracted from the text string of the answer option using the regular expression *e.target.value.match(/d+/)*. The extracted value is then converted to an integer using *parseInt()* before being stored in the state variable *selected*. The set of confirmed answers forms the answer vector:

$$A = (a_1, a_2, \dots, a_{21}), \quad a_i \in \{0, 1, 2, 3\}. \quad (4)$$

Although there are 21 diagnostic questions ($i = 1, \dots, 21$), the zero-based implementation index k is defined as $k = i - 1$. Thus, q_k corresponds to q_i under the relation $i = k + 1$. Therefore, $k \in \{0, 1, \dots, 20\}$ corresponds to the diagnostic questions, while $k = 21$ is not associated with any question but instead serves as a termination state indicating that the test has been completed. The state variable *activeQuestion* starts at 0 and is incremented by 1 after each valid answer until it reaches 21, which triggers the end of the testing process.

The total score S is calculated as the sum of the response coefficients:

$$S = \sum_{i=1}^{21} a_i, \quad S \in [0, 63]. \quad (5)$$

In the software implementation, this corresponds to *answers.reduce((sum, value) => sum + value, 0)*, the result of which is stored in the variable *anw*, representing S .

Based on the obtained value S , the system generates a diagnostic conclusion D according to the classification function, precisely aligned with the threshold conditions of the ternary operators in the code:

$$D(S) = \begin{cases} d_1, & S < 9, \\ d_2, & 9 \leq S < 18, \\ d_3, & 18 \leq S < 29, \\ d_4, & S \geq 29, \end{cases} \quad (6)$$

where the values of the diagnostic findings are defined as follows:

$d_1(anw1)$ – normal condition, no symptoms of depression;

$d_2(anw2)$ – mild to moderate depression;

$d_3(anw3)$ – moderately severe depression;

$d_4(anw4)$ – severe depression.

In the software implementation, the corresponding diagnostic conclusions are defined as constant string variables $anw1$, $anw2$, $anw3$, and $anw4$.

The use of strict inequalities in the formula ($S < 9$, $S < 18$, $S < 29$) corresponds to the comparison operators $<$ in the code: $(anw < 9) ? anw1 : (anw < 18) ? anw2 : (anw < 29) ? anw3 : anw4$, which ensures unambiguous and correct classification for any integer value of $S \in [0, 63]$.

The overall decision-making logic of the system can be represented as a mapping:

$$f: A \rightarrow D, f(a) = D\left(\sum_{i=1}^{21} a_i\right), \quad (7)$$

that is, the system performs a unique mapping of the user's response vector to the set of diagnostic classes $\{d_1, d_2, d_3, d_4\}$.

The correctness of the mapping $f: A \rightarrow D$ is guaranteed by construction, since the classification function exhaustively partitions the entire range of possible scores into mutually exclusive and collectively exhaustive intervals.

To ensure the correctness of the formation of the vector A , the condition of completeness of input is defined – a mandatory requirement according to which the transition to the question q_{i+1} is possible only if:

$$a_i \neq \emptyset, \forall i = 1, \dots, 21.$$

where $\emptyset$ denotes the absence of a selected response. In the implementation, the condition $a_i \neq \emptyset$ corresponds to $selected \neq ''$.

In the software implementation, this condition is controlled in the *clickHandler* event handler through a *switch(selected)* construct: if *selected = ''*, the system sets an error message via *setError('* You need to choose one answer option!')* and does not proceed. Otherwise, *setActiveQuestion(activeQuestion + 1)* is called.

The usage scenarios of the system are formally described by the set:

$$UC = \{uc_1, uc_2, uc_3, uc_4, uc_5\}, \quad (8)$$

where uc_1 – review of the test description;

uc_2 – start testing (*onClick={() => setTest(true)}*);

uc_3 – answer to question q_i (*onChange={changeHandler}*);

uc_4 – moving to the next question q_{i+1} with validation $a_i \neq \emptyset$ (*onClick={clickHandler}*);

uc_5 – receiving a diagnostic conclusion $D(S)$.

The uc_4 use case includes a mandatory validation subprocess:

$$uc_4 \xrightarrow{\text{include}} val(a_i \neq \emptyset), \quad (9)$$

which makes it impossible to form an incomplete response vector A .

The architecture is defined as a three-component structure:

$$Sys = \langle App, ExpertTest, data \rangle,$$

where the *App* component controls the binary state $v \in \{0, 1\}$ (*visibleTest*):

$$View(v) = \begin{cases} \text{introductory screen}, & v = 0; \\ ExpertTest, & v = 1. \end{cases} \quad (10)$$

The *ExpertTest* component manages the internal state:

$$\sigma = (k, \hat{a}, A, e), \quad (11)$$

where k – current question index (*activeQuestion*);

$\hat{a}$ – the currently selected option (*selected*), not yet added to the vector;

A – accumulated vector of confirmed responses (*answers*);

e – (*error*) message status.

The appearance of the *title: 'end'* marker at $k = 21$ initiates the calculation of S and the call of the function $D(S)$.

The dynamics of the system's operation are described by a state diagram with the set:

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}, \quad (12)$$

where s_1 – introductory screen (*visibleTest = false*);

s_2 – displaying a question q_i ;

s_3 – answer selected (*selected \neq ''*);

s_4 – validation error (*error \neq ''*);

s_5 – status update (*answers, activeQuestion*);

s_6 – representation of the result $D(S)$.

Transitions between states are determined by the following conditions:

$$\begin{aligned} s_1 &\xrightarrow{v=1} s_2 \xrightarrow{\hat{a} \neq \emptyset} s_3 \xrightarrow{\text{clickHandler}} s_5 \xrightarrow{k < 21} s_2 \\ s_3 &\xrightarrow{\hat{a} \neq \emptyset} s_4 \xrightarrow{\text{changeHandler}} s_3 \\ s_5 &\xrightarrow{k=21} s_6. \end{aligned}$$

where the transitions between s_2 , s_3 and s_4 use $\hat{a}$ (*selected*), since at this stage the answer has not yet been confirmed and added to A.

Thus, the implementation corresponds to the formal model: the set Q is represented by *data.quest*, the vector A by *answers*, and the function D(S) is implemented via conditional operators. The input completeness condition $a_i \neq \emptyset$ is enforced in the event handler.

3 Development of the Architecture and Operation Algorithms of an Expert System Based on the Beck Scale

3.1 Software System Architecture

The proposed solution is based on a three-component architecture $Sys = \langle App, ExpertTest, data \rangle$, which reflects the structure of the formal model and its implementation in a web environment (Figure 1).

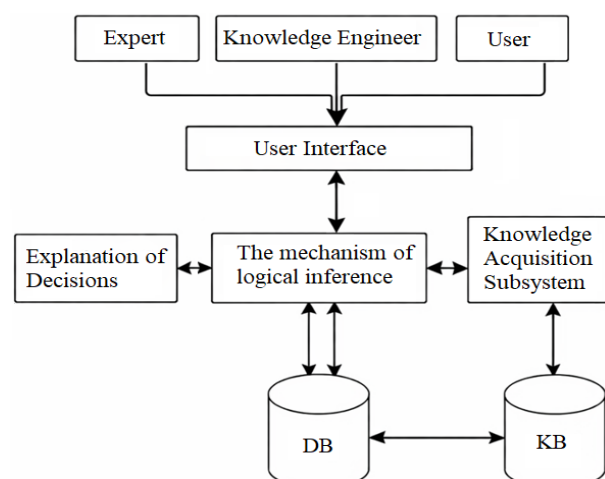


Fig. 1: Standardized architecture of an expert system
Source: created by the authors

The user interface is implemented using HTML5 and CSS3; however, these technologies serve only a supporting role. The core functionality of the system

is determined by the formal model and the inference logic implemented in the application components, while the interface layer is responsible solely for structured data input and result presentation.

This ensures that the diagnostic tool is accessible to a wide range of users, regardless of the software they choose. The system's introductory screen is shown in Figure 2, while the contact information panel and the test initiation interface are presented in Figure 3.

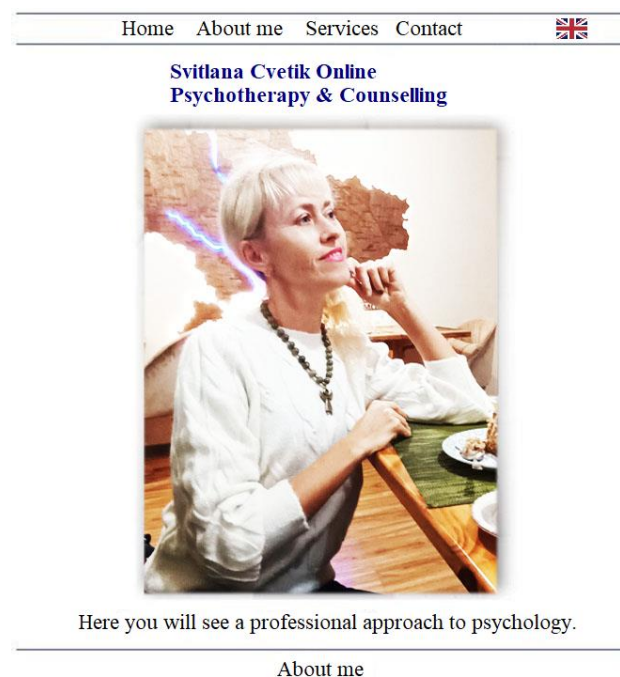


Fig. 2: Introductory screen of the web-based expert system

Source: created by the authors

The knowledge base contains formalized knowledge about depression in the form of the 21-item Beck Depression Inventory, with four response options for each item, which corresponds to the structure of the *data.quests* dataset.

The database, or the system's working memory, stores the current state of the test: the user's answer vector *answers* and the index of the current question *activeQuestion*.

The inference engine is a software module that calculates the total score S and applies the classification function D(S) to generate a diagnostic conclusion.

In this implementation, the explanation subsystem is presented in the form of displaying the textual test result with the option to contact a specialist. The output of the result is accompanied by the specialist's contact details (phone, Telegram, WhatsApp), which are available on the introductory screen. This implements feedback – the user does

not just receive a diagnosis but also sees where to turn for help.

- Psychological trauma
- Breakups
- Loss

Consultations

- Individual
- Family
- Couples
- Group

Consultations online and in-office at 22 Kosmicheskaya Street, Kharkiv.

Call

WhatsApp

Telegram

Contacts

☎ Mobile: +38(057) 702 14 86

✉ Telegram +38(057) 702 14 86

💬 WhatsApp +38(057) 702 14 86

Cost per consultation: 500 UAH.

Want to take a test and find out the result?

This questionnaire consists of 21 groups of statements.

Please read each group of statements carefully, then choose one statement from each group that best describes how you have felt during the past week, including today.

Take the test

Tel: +38(057) 702 14 86

© Cvetik Psychology

Fig. 3: Specialist contact information and test initiation interface

Source: created by the authors

The knowledge acquisition subsystem is implemented indirectly through the ability to edit the *data.quests* array when it is necessary to expand or modify the knowledge base.

This architecture directly implements the formal structure defined in Section 2, where the system is represented as $Sys = \langle App, ExpertTest, data \rangle$.

The *App* component manages the global system state v (*visibleTest*) and controls the transition between the introductory screen and the testing process in accordance with the view function (Formula 10).

The *ExpertTest* component implements the inference mechanism by operating on the internal state σ (Formula 11), including the formation of the response vector and the execution of the classification function.

The *data* module represents the knowledge base Q as a static structure used during the diagnostic process.

3.2 Software Implementation of the Knowledge Base

The formal model of the knowledge base, represented as the set $Q = \{q_1, q_2, \dots, q_{21}\}$, along with its mathematical description, is thoroughly defined in Section 2. In this subsection, we discuss

the actual software implementation of this model within the system's code.

The set Q (Formula 1) is implemented as an array of objects *data.quests*, where each object corresponds to a question q_i (Formula 2) and has a *title* field containing the text of the symptom t_i , and a *choices* array with four answer options, corresponding to the set C_i (Formula 3). The basic logic of the expert diagnostic system, implemented within the *ExpertTest* component, is shown in Figure 4.

```
import React, { useState } from 'react';
import { useRef } from 'react';
import './test.css'

import { data } from './data';

import React, { useState } from 'react';
import { useRef } from 'react';
import './test.css'

import { data } from './data';

export default function ExpertTest() {

  const [activeQuestion, setActiveQuestion] = useState(0);
  const [selected, setSelected] = useState("");
  const [error, setError] = useState("");
  const [answers, setAnswers] = useState([]);
  const radiosWrapper = useRef();

  const changeHandler = (e) => {
    setSelected(parseInt(e.target.value.match(/\d+/)));
    if(error) {
      setError("");
    }
  }

  const anw1 = 'Normal state.'
  const anw2 = 'Mild to moderate depression.'
  const anw3 = 'Moderately severe depression.'
  const anw4 = 'Severe depression.'

  const anw = answers.reduce((sum, value) => sum + value, 0);

  let message = (anw < 9) ? anw1 :
    (anw < 18) ? anw2 :
    (anw < 29) ? anw3 :
    anw4;

  console.log(message);
}
```

Fig. 4: The Basic Logic of the Expert Diagnostic System (Code Snippet)

Source: created by the authors

A fundamental feature of the implementation is that the diagnostic coefficient $a_i \in \{0, 1, 2, 3\}$ is not stored as a separate field but is embedded directly into the text string of the response option as its first

character – for example, «0 - I do not feel sad» or «3 - I am so sad ...». This approach allows the knowledge base to be stored as a single, compact, static data structure, simplifies code maintenance, and ensures consistency between the text displayed to the user and the diagnostic value processed by the system.

During testing, the numerical value of the coefficient is extracted from the string using the regular expression *e.target.value.match(/\d+/)* in the *changeHandler*. This extracts the first digit sequence in the string – the diagnostic coefficient, which, according to the knowledge base structure, is always located at the beginning of the answer option string. This ensures the correct extraction of the coefficient itself, rather than other numbers that may be contained in the answer text – for example, «2 - I wake up 1-2 hours earlier than usual...».

The last element of the *data.questions* array contains the end marker { *title: 'end'* } and is not a diagnostic question. Its appearance at the *data.questions[activeQuestion]* position serves as a signal for the *ExpertTest* component to complete the collection of answers, calculate S, and call the function D(S).

Thus, the *data.questions* array contains 22 elements: 21 diagnostic questions corresponding to the set Q, and one special end marker that does not affect the calculation of the diagnostic indicator but is used to manage the testing logic.

3.3 System Operation Algorithm

The system's operating algorithm implements the sequence of transitions between states S, defined in Section 2 (Formula 12), and is shown in Figure 5.

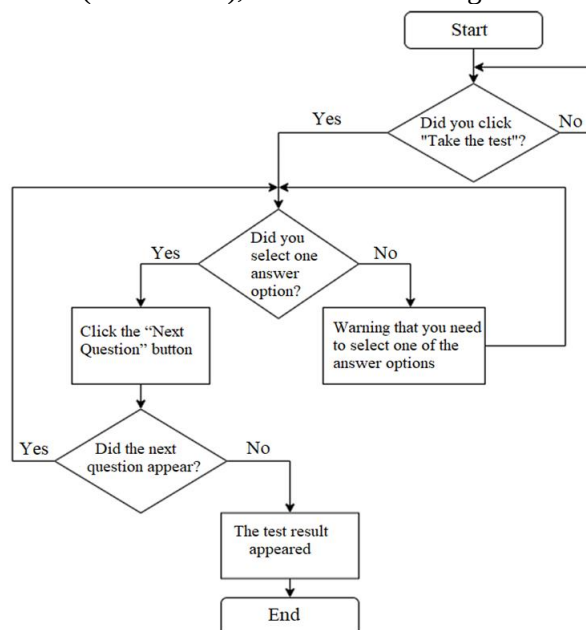


Fig. 5: System operation algorithm
Source: created by the authors

When the page loads, the initial system state is set: *visibleTest = false*, *activeQuestion = 0*, *answers = []*, *selected = ''*, *error = ''*. The user is on the introductory screen, which displays the test description and a start button. The system waits for the «Start Test» button to be clicked and remains in the initial state until then.

When the button is clicked, *setTest(true)* is called, which sets *visibleTest = true* and switches the system to testing mode, initiating the transition from state *s₁* to state *s₂*.

In state *s₂*, the current question *q_k* is retrieved from the *data.questions[activeQuestion]* array, and the corresponding set of answer options *C_k* is rendered as radio buttons. Initially, *k = 0*, which corresponds to the first question *q₁* under one-based indexing. Here, *C_k* denotes the set of answer options for the current question.

Next, the system checks whether the user has selected an answer. If no option is selected, the system remains in a waiting state. Once the user selects an option, the *changeHandler* is triggered. This handler extracts a numerical coefficient from the text string of the selected option using the regular expression *e.target.value.match(/\d+/)*. This expression captures the first sequence of digits – the diagnostic coefficient, which, according to the knowledge base structure, is always located at the beginning of the answer option string. The extracted value is stored in the state variable *selected* as a temporarily chosen option *â*, corresponding to the system's transition from state *s₂* to state *s₃*. At this stage *â* is not yet part of the vector A and has not been added to the *answers* array.

After the user selects an answer option, they click the «Next Question» button. This triggers the *clickHandler*, which verifies that an answer has been selected, as specified in Section 2. The verification is performed using a *switch(selected)* statement.

If *selected = ''* – the condition is not met and the process transitions to state *s₂*: it displays a warning using *setError('* You must select one answer option!')*, indicating that an option must be chosen, and the transition to the next question is blocked.

The user returns to selecting an answer, which corresponds to the transition *s₄ → s₃*.

If *selected ≠ ''* – the condition is met and the system transitions to state *s₅*: the *selected* value is added to the *answers* array via *setAnswers([...answers, selected])*, the *selected* variable is reset to an empty string, the error message is cleared, and the *activeQuestion* counter

is incremented by 1 via $setActiveQuestion(activeQuestion + 1)$.

After the state update, the system checks whether the next question has appeared, i.e., whether the value of $data.questions[activeQuestion].title$ differs from the 'end' completion marker.

If so, the transition $s_5 \rightarrow s_2$ is executed, and the system returns to displaying the next question q_{k+1} . This sequence of steps is repeated for each of the 21 questions on the Beck scale.

If the value equals 'end', there is no next question, and the transition $s_5 \rightarrow s_6$ is performed. This means that all 21 questions have received confirmed answers, and the vector A (Formula 4) is fully formed.

In state s_6 , the system calculates the total score S and generates the conclusion $D(S)$ according to the mathematical model defined in Section 2 (Formulas 5 and 6).

In the software implementation, the calculation of S corresponds to the expression $answers.reduce((sum, value) => sum + value, 0)$, the result of which is stored in the variable anw .

The $D(S)$ classification function is implemented through a chain of ternary operators ($anw < 9$) ? $anw1$: ($anw < 18$) ? $anw2$: ($anw < 29$) ? $anw3$: $anw4$, which returns one of the diagnostic conclusions $d_1...d_4$. The result is displayed to the user as a message: «*You have completed the test!*» along with the corresponding diagnostic conclusion $D(S)$, which completes the mapping $f: A \rightarrow D$ (Formula 7), after which the algorithm terminates.

Thus, the implemented algorithm accurately reproduces the logic of the flowchart (Figure 5) and ensures a deterministic, sequential, and fully validated procedure for collecting diagnostic data. This guarantees the correct formation of the answer vector A and the unambiguous computation of the diagnostic conclusion $f(A) = D(S)$ in accordance with the formal specification defined in Section 2.

4 Research Results and Discussion

In this study, a web-oriented expert system for diagnosing depression based on the Beck Depression Inventory was developed. To verify the system's functionality and confirm its compliance with the formal model defined in Section 2, a series of test runs was conducted using various response scenarios.

The system home screen, featuring the test description and the «Start Test» button, is shown in Figure 6.

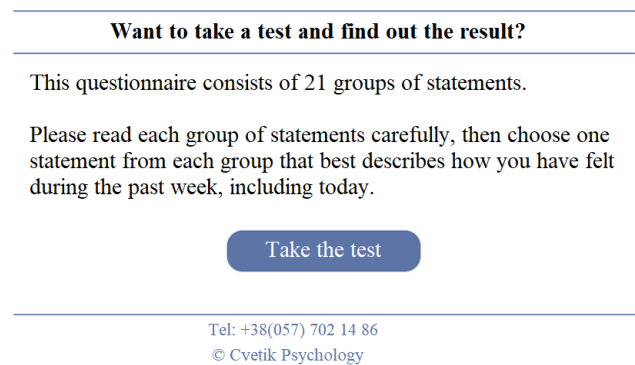


Fig. 6: System home screen

Source: created by the authors

The user interface, built using HTML5 and CSS3, provides intuitive navigation and displays correctly on various devices.

The introductory screen contains information about the specialist, a list of services, and contact details, allowing users to seek professional help immediately after receiving their results.

The sequence of 21 questions displayed by the system, each accompanied by four answer options with diagnostic coefficients ranging from 0 to 3, is shown in Figure 7.

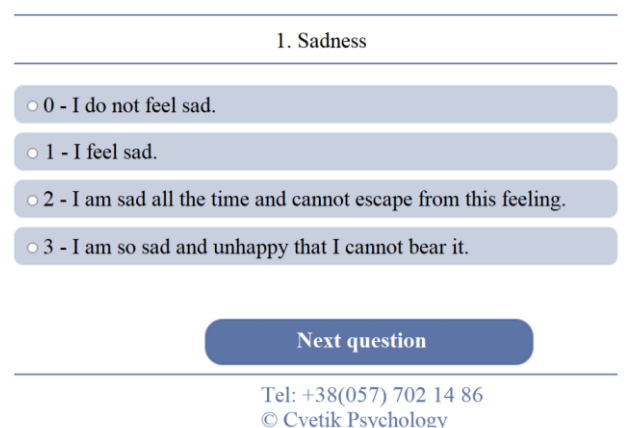


Fig. 7: Question interface with four answer options and diagnostic coefficients (0–3)

Source: created by the authors

The visual presentation of questions using radio buttons ensures ease of selection and prevents multiple choices for a single question.

During testing, the validation mechanism implemented in the *clickHandler* was verified.

When attempting to proceed to the next question without selecting an answer, the system correctly blocks the transition and displays the message «* You must select an answer option!» as shown in Figure 8. This guarantees the fulfillment of the input completeness condition $a_i \neq \emptyset$ for all 21 questions

and ensures the correct formation of the answer vector A .

Fig. 8: Validation message blocking transition when no answer is selected

Source: created by the authors

After the test is completed, the system calculates the total diagnostic score (Formula 5) and applies the classification function $D(S)$ according to Formula 6.

To verify the correctness of the algorithm, four response scenarios corresponding to different diagnostic categories were simulated. In the first scenario, the user selected options with the minimum coefficient of 0 for all questions, which yielded $S = 0$ and the conclusion: «Normal state, no symptoms of depression».

In the second scenario, answers were chosen with coefficients that yielded a sum of $S = 13$, and the system output the conclusion: «Mild to moderate depression». The third scenario, with answers that gave a sum of $S = 23$, produced the result: «Moderately severe depression». Finally, the fourth scenario, with maximum scores of 3 for all questions, yielded a sum of $S = 63$ and the conclusion «severe depression», as shown in Figure 9. In all cases, the results fully corresponded to the expected values according to the Beck Scale methodology, which confirms the correct implementation of the classification function $f: A \rightarrow D$.

Fig. 9: Test result for the maximum score scenario ($S = 63$)

Source: created by the authors

Special attention was paid to testing the boundary values of the diagnostic intervals. The testing confirmed that with a score of $S = 8$, the system outputs the conclusion «normal state»; with $S = 9$ – «Mild to moderate depression»; with $S = 17$ – also «Mild to moderate depression»; with $S = 18$ – «Moderate to severe depression»; with $S = 28$ – «Moderate to severe depression»; and with $S = 29$ – «Severe depression». This demonstrates the precise execution of the ternary operator conditions $(anw < 9) ? anw1 : (anw < 18) ? anw2 : (anw < 29) ? anw3 : anw4$ and confirms the correct implementation of the classification function $D(S)$ at the boundary values of the diagnostic intervals.

An important aspect of the system's operation is its integration with the specialist's information page. After receiving the test result, the user has the opportunity to immediately contact a psychologist via the phone numbers provided on the introductory screen (Telephone, Telegram, WhatsApp). This architecture aligns with the recommendations for creating screening diagnostic tools that not only provide information about one's condition but also offer a pathway to professional help.

The conducted testing has confirmed that the developed system fully conforms to the formal model described in Section 2 and implements all the defined use cases: $UC = \{uc_1, uc_2, uc_3, uc_4, uc_5\}$. The system ensures a deterministic mapping of the answer vector onto the set of diagnostic classes $\{d_1, d_2, d_3, d_4\}$, guarantees the completeness of data input, and provides the user with an intuitive interface for self-administration of the test.

The developed system is intended solely for screening and does not replace a comprehensive clinical interview. The result should be considered a preliminary indication that may guide users toward seeking professional help. The system does not take into account the patient's individual characteristics, comorbid conditions, or other factors that may affect mental health; therefore, a final diagnosis must be made by a qualified psychiatrist or psychotherapist.

Thus, the results of the study confirm the functionality and effectiveness of the proposed web-based expert system as a tool for the automated screening and diagnosis of depressive disorders. The system can be used in the clinical practice of psychologists for the preliminary assessment of patients' conditions, as well as a standalone tool for self-diagnosis, with a subsequent recommendation to consult a specialist.

The results are in line with those of [7], which compared the psychometric properties of the paper and online versions of the Beck Depression

Inventory. That study revealed a high correlation between the results obtained using the paper and online formats ($r = .89$), confirming the reliability of the questionnaire's electronic administration. This corresponds to the correct functioning of the developed system, where test results for various response scenarios fully correspond to the theoretically expected values on the Beck scale.

In [8], conducted on a sample of over 58,000 individuals, it was also confirmed that instruments based on Beck's questionnaires are reliable for online depression screening. Using modern analytical methods, including Item Response Theory, the researchers confirmed the high accuracy of measurements in a digital environment. This aligns with the logic of our system, which combines responses to all 21 questions to form a single diagnostic conclusion.

In contrast to the aforementioned studies, which are primarily psychometric in nature, this work focuses on the software-architectural implementation. The main distinction of our approach lies in representing the diagnostic process as a three-component expert system $Sys = \langle App, ExpertTest, data \rangle$, which allows not merely to collect data but to automate decision-making logic at the software code level. While psychometric studies demonstrate that online versions of the Beck Scale work, our work shows how exactly such a system can be technically implemented, ensuring a deterministic mapping $f: A \rightarrow D$ in accordance with the formal model defined in Section 2.

While [8] treats the web environment merely as a means of conducting surveys, our study focuses on formalizing the knowledge base and ensuring data integrity through validation algorithms (Equation 9). The use of a component-oriented approach (React.js) ensures high-speed processing of the response vector directly on the client side, which enhances the confidentiality and interactivity of the process.

A distinctive feature of the developed system is the integration of the diagnostic module with the feedback subsystem. This transforms the system from a passive screening tool into an active element of clinical communication, providing the user not only with a diagnosis but also with an immediate algorithm for further action.

To clearly define the scientific novelty and engineering features of the developed solution, a comparative analysis was conducted with current research in this field (Table 1).

Table 1. Comparative analysis of the proposed expert system with existing studies

Criterion of comparison	Holländare et al. (2010) [7]	Salcedo-Callado et al. (2025) [8]	Proposed System
Primary focus	Psychometric properties (Paper vs Web)	Large-scale validation (IRT analysis)	Software architecture and automation
Research goal	To prove format equivalence	To verify measurement accuracy	To develop an expert diagnostic system
Formalization method	Statistical correlation	Item Response Theory (IRT)	Set theory and algorithms
Technology	Generic web interface	Online survey platform	React.js component-based structure

Source: created by the authors

Unlike previous studies, which focused on the psychological validation of the Beck Scale in digital formats, the proposed system introduces a formalized engineering approach. The key difference lies in the transition from a simple web survey to a structured expert system with a three-tier architecture, automated data validation algorithms, and integrated decision-making logic.

5 Discussion on Mathematical Limitations and Extensions

The proposed model is deterministic and based on a discrete threshold classification function. While this ensures interpretability and reproducibility, it does not account for uncertainty in user responses. Future improvements may include probabilistic extensions using Bayesian inference or fuzzy logic to model uncertainty in symptom interpretation.

Additionally, the current classification function can be analyzed in terms of sensitivity to boundary values, particularly near threshold transitions ($S = 9, 18, 29$), where small changes in responses may lead to different diagnostic outcomes. As shown in experimental results (Section 4), the system correctly handles these boundary cases ($S = 8, 9, 17, 18, 28, 29$), confirming precise threshold implementation.

The proposed deterministic classification scheme demonstrates high robustness to small variations in input values, except at predefined threshold boundaries, where sensitivity is confirmed through boundary value testing.

From a computational perspective, the algorithm operates in linear time $O(n)$, where $n = 21$ is the

number of questions, with constant auxiliary space complexity $O(1)$, since only scalar accumulators and a response array are used. This ensures efficient execution in web environments.

In comparison with alternative approaches – such as rule-based inference engines, Bayesian networks, or machine learning-based classifiers (e.g., logistic regression, decision trees, neural networks) – the proposed deterministic model offers greater transparency and ease of validation. However, future work could explore hybrid models that combine deterministic thresholds with probabilistic inference to improve robustness in ambiguous cases.

6 Conclusion

This study focused on the development and investigation of a web-based expert system designed to diagnose depression using the Beck Scale. Unlike existing psychometric studies, which focus on validating online versions of questionnaires, the proposed system is viewed as a software-architectural solution that automates the logic of diagnostic decision-making at the level of program code.

The following main results were obtained in the course of the work.

Formalization of a model for diagnosing depressive disorders. This paper proposes a formal formulation of the diagnostic problem based on the Beck Scale. It defines a set of questions, the structure of the knowledge base, the user response vector, and a classification function that maps the total score to a set of diagnostic conclusions. The resulting model provides a mathematical foundation for the algorithmic implementation of automated diagnosis within a web-based system.

A three-component architecture based on React.js is proposed: the root *App* component manages the switching between the introductory screen and the testing process, the *ExpertTest* component implements the expert system logic by operating on its internal state, and the static *data* module provides storage for the knowledge base. This approach allows for the integration of diagnostic decision support algorithms into a web environment with a clear separation of component responsibilities.

This paper describes the implementation of a knowledge base in the form of an array of *data.quests* objects, where diagnostic coefficients are embedded directly into the text strings of the answer options, ensuring a compact and consistent data structure. It also provides a detailed description of the system's operational algorithm, which

implements a sequence of transitions between states in accordance with a formal model. The algorithm includes validation of the answer selection, updating the answer vector, and calculating the diagnostic conclusion through a chain of ternary operators. The chosen technology stack (React.js, HTML5, CSS3) ensures high data processing speed on the client side, interactivity, and a user-friendly interface.

The developed system was tested using four response scenarios corresponding to different diagnostic categories. The testing confirmed the correct operation of the algorithms for all diagnostic intervals, including boundary values. The system ensures that all required fields are filled out through a validation mechanism and provides instant feedback to the user. Integration with the specialist's contact information creates a continuous chain of «diagnosis – recommendation – request for assistance», which enhances the system's practical value.

The novelty of this work lies in the following:

1. The formalization of the task of diagnosing depression based on the Beck Scale as a deterministic mapping of a response vector to a set of diagnostic classes, adapted to the architecture of a web-based expert system built on React.js.

2. The development of a three-component architecture that combines a static knowledge base, a dynamic testing state, and decision-making logic into a single software structure.

3. The creation of a formal model of system states with clearly defined transitions, ensuring deterministic program behavior and correct processing of all possible user actions.

4. Implementation of an original mechanism for embedding diagnostic coefficients directly into the text strings of answer options, followed by extraction via a regular expression, which allows the knowledge base to be stored as a compact static structure without data duplication.

The practical significance of the results lies in the possibility of implementing the developed system into psychologists' daily practice for preliminary screening of patients' conditions, as well as serving as a standalone tool for self-diagnosis, followed by a recommendation to consult a specialist. Integrating the system with a specialist's website creates a unified information space, which increases the accessibility of psychological assistance.

Areas for further research include expanding the knowledge base with other psychometric scales (in particular, the Spielberger Anxiety Scale), adding a

feature to save result history to track changes in the user's condition, developing an administrative panel for the specialist with the ability to view anonymized statistics, and integration with remote counseling systems.

Thus, the research objective has been achieved: the web-based expert system developed automates the diagnosis of depressive disorders using the Beck Scale, thereby increasing the accessibility of screening diagnostics and creating conditions for users to seek professional help promptly.

This manuscript is original, has not been published previously, and is not currently under consideration for publication elsewhere. Any overlap with prior work is limited to general background concepts and is properly cited. The core methodology, formal model, and implementation presented in this paper are novel.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] D. N. Klein, Diagnosis and classification of depressive disorders, *APA handbook of depression: Classification, co-occurring conditions, and etiological processes*, 2026, Vol. 1, pp. 21–46. DOI: 10.1037/0000437-002.
- [2] A. Cipriani, K. B. P. Fernandes, B. H. Mulsant, O. Efthimiou, N. Williams, S. Mort, E. G. Ostinelli, A decision-support system to personalize antidepressant treatment in major depressive disorder: a randomized clinical trial, *JAMA*, 2026, Vol. 335, Issue 14, pp. 1219–1231. DOI: 10.1001/jama.2026.1327.
- [3] L. M. Brooks, B. M. Brooks, A. S. Arp, C. R. Dove, L. C. Rogers, R. Michel, V. Clinton, J. Labovitz, D. G. Armstrong, Understanding DREADD (Diabetes-Related Extremity Amputation Depression and Distress), *Clinics in Podiatric Medicine and Surgery*, 2026, Vol. 38, Issue 1, pp. 94–100. DOI: 10.1053/j.semvasc Surg.2025.01.002.
- [4] N. Shusharina, D. Yukhnenko, S. Botman, V. V. Savinov, G. Kamyshev, D. Sayapin, I. Voznyuk, Modern methods of diagnostics and treatment of neurodegenerative diseases and depression, *Diagnostics*, 2023, Vol. 13, Issue 3, 573, pp. 1–27. DOI: 10.3390/diagnostics13030573.
- [5] A. Taran, S. Sotnik, Low-Code/No-Code Web Platforms: Opportunities and Limitations, in Proceedings of the All-Ukrainian Conference "Intelligent Civil Safety Technologies and Robotic Systems for Emergency and Rescue Operations" (ICSTRO-2026), Kharkiv, Ukraine, February 12–13, 2026, pp. 96–100.
- [6] A. Andreiev, S. Sotnik, Web Application Security: Protection Against Modern Cyber Threats: Overview of Key Vulnerabilities (XSS, CSRF, SQL Injections), Protection Methods, Use of HTTPS, Authentication, and Authorization, in Proceedings of the IX International Conference "Manufacturing & Mechatronic Systems 2025", Kharkiv, Ukraine, October 25–26, 2025, pp. 66–70.
- [7] F. Holländare, G. Andersson, I. Engström, A comparison of psychometric properties between Internet and paper versions of two depression instruments (BDI-II and MADRS-S) administered to clinic patients, *Journal of Medical Internet Research*, 2010, Vol. 12, Issue 5, pp. e49. DOI: 10.2196/jmir.1392.
- [8] T. Salcedo-Callado, N. Hernández-Llanes, R. Sánchez-Domínguez, R. Saracco-Alvarez, R. Marín-Navarrete, IRT analysis of the BDI-II for early online depression detection: validation in a Mexican population, *Frontiers in Psychology*, 2025, Vol. 16, pp. 1–10. DOI: 10.3389/fpsyg.2025.1562016.
- [9] J. R. Evans, A. Mathur, The value of online surveys: a look back and a look ahead, *Internet Research*, 2018, Vol. 28, Issue 4, pp. 854–887. DOI: 10.1108/IntR-03-2018-0089.
- [10] L. Hohwü, H. Lyshol, M. Gissler, S. H. Jonsson, M. Petzold, C. Obel, Web-based versus traditional paper questionnaires: A mixed-mode survey with a nordic perspective, *Journal of Medical Internet Research*, 2013, Vol. 15, Issue 8, pp. e173. DOI: 10.2196/jmir.2595.
- [11] J. Flanagan, C. T. Beck, Polit & Beck's nursing research: Generating and assessing evidence for nursing practice, *Lippincott Williams & Wilkins*, 2024, 832 p.
- [12] Psychology Test Platform, [Online]. <https://psytests.be> (Accessed Date: March 14, 2026).
- [13] Mental Health Online. Expert programs, information and resources to help you manage

your mental wellbeing, [Online].
<https://www.mentalhealthonline.org.au>
(Accessed Date: March 14, 2026).

- [14] Qualtrics. Customer Experience Intelligence_Understand every customer. Act when it counts, [Online].
<https://www.qualtrics.com> (Accessed Date: March 14, 2026).
- [15] Rozumka, [Online]. <https://rozumka.ua>
(Accessed Date: March 14, 2026).
- [16] Mamook, [Online]. <https://momook.com>
(Accessed Date: March 14, 2026).
- [17] M. P. Grosz, G. Olaru, M. D. Back, J. F. Rauthmann, Assessing the ABCDs of Machiavellianism: Development and Validation of the M4 Scale Using Ant Colony Optimization, *Collabra: Psychology*, 2025, Vol. 11, Issue 1, 142730. pp. 1–22. DOI: 10.1525/collabra.142730.
- [18] C. Metcalfe, T. Wilson, Comparability of YouGov and Qualtrics samples in survey experiments: A short report, *Journal of Experimental Criminology*, 2025, pp. 1–17. DOI: 10.1007/s11292-025-09701-9.
- [19] M. S. W. Yacoub, A. Nicholas, Learning Modules Created on Qualtrics: An Adaptable Tool for Student Engagement, *The American Biology Teacher*, 2025, Vol. 87, Issue 8, pp. 482–485. DOI: 10.1525/abt.2025.87.8.482.
- [20] Y. Desnelita, M. Cesar, G. Gustientiedina, A. Hajjah, R. N. Putri, Implementation of certainty factor method in mental health diagnosis expert system in adolescents aged 18–24 years, *Journal of Applied Business and Technology*, 2025, Vol. 6, Issue 1, pp. 42–51. DOI: 10.35145/jabt.v6i1.195SD.
- [21] S. Amatya, E. Sage, D. Edgerton, Guidance for eCOA Development in Clinical Trials, *Journal of the Society for Clinical Data Management*, 2025, 4, pp. 1–23. DOI: 10.47912/jscdm.376.
- [22] S. J. Kho, S. Diamond, P. Lafeuille, J. H. Schatz, F. Maura, J. Aptekar, F. L. Locke, Racial and clinical determinants of response in 2304 large B-cell lymphomas treated with CD19 CAR T in clinical trials, *Blood advances*, 2026, Vol. 10, Issue 5, pp. 1497–1507. DOI: 10.1182/bloodadvances.2025016361.
- [23] E. Abu Bonsra, P. Y. Atsrim, A. Korankye, J. Komesuor, Experiences and coping strategies among patients with Chronic Renal Failure (CRF) in Ghana: A phenomenological study, *PLOS Mental Health*, 2025, 2.9: e0000279, pp. 1–22. DOI: 10.1371/journal.pmen.0000279.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in present research, at all stages from the formulation of problem to final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

Self-financing.

Conflict of Interest

No conflicts of interest in this research.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US