# Personality Test Web Application: A Bachelor Project Report

**Author:** Shaghayegh Shafiee
**Supervisor:** Dr. Behshid Behkamal
**Institution:** Ferdowsi University of Mashhad, Faculty of Engineering, Department of Computer Engineering
**Degree:** Bachelor of Science in Computer Engineering
**Date:** [March 9th 2025]

# Table of Contents

# 1. Introduction

## 1.1 Project Overview

This project involves the development of a web-based personality test application designed to cater to both adults and children. The application dynamically adjusts its interface and question presentation based on the selected mode—adult or child. It incorporates voice recognition technology, utilizing the Whisper model, to allow users to answer questions verbally, thereby enhancing accessibility and user experience. Built using Flask for the backend and HTML/CSS/JavaScript for the frontend, the application reads questions from Excel files and stores user responses in a similar format. This project was completed as part of my Bachelor of Science in Computer Engineering at Ferdowsi University of Mashhad, under the supervision of Dr. Behashid Behkamal.

## 1.2 Objectives

The primary goals of this project were:

- To create an interactive and user-friendly platform for personality assessment.
- To implement a system that adapts to
- different user modes (adult and child) with appropriate question sets.
- To integrate voice recognition technology for seamless user interaction.
- To ensure the application is robust, handles errors gracefully, and provides a smooth user experience.

## 1.3 Scope

The scope of this project includes the development of a web application using Flask for the backend and HTML/CSS/JavaScript for the frontend. It involves integrating the Whisper model for audio transcription and using Excel files to store questions and user responses. The application is designed to run on a local server and has been tested with specific question sets for both adult and child modes. Future enhancements, such as cloud deployment or additional personality assessment features, are beyond the current scope.

# 2. Background and Literature Review

## 2.1 Personality Assessment Methods

Personality assessments are widely used in psychology to evaluate individual differences in behavior, emotion, and cognition. Common methods include self-report questionnaires, projective tests, and behavioral observations. This project adopts a self-report questionnaire approach, where users respond to a series of questions to reveal aspects of their personality. While it does not strictly adhere to established frameworks like the Big Five or Myers-Briggs, it provides a flexible foundation for such adaptations.

### 2.2 Web Application Development with Flask

Flask is a lightweight web framework for Python, ideal for rapid development of web applications. It was chosen for this project due to its simplicity, flexibility, and seamless integration with Python libraries such as Pandas for data handling and Whisper for audio transcription. Flask's microservices architecture supports the modular design of this application.

### 2.3 Audio Transcription with Whisper

Whisper, developed by OpenAI, is an automatic speech recognition (ASR) system capable of transcribing spoken language into text with high accuracy. In this project, Whisper transcribes user responses recorded through the web interface, enabling voice-based interaction and broadening the application's accessibility.

## 3. System Design and Architecture

### 3.1 System Architecture

The application follows a client-server architecture. The server, built with Flask, handles client requests, processes data, and interacts with external services like the Whisper model. The client-side, developed with HTML, CSS, and JavaScript, provides the user interface and manages user interactions.

### 3.2 Database Design (Excel Files)

Instead of a traditional database, Excel files store questions and user responses due to their simplicity and ease of management during development. Personality_Test.xlsx contains adult questions, while Child_Mode_Questions.xlsx holds child questions. Each row represents a question with associated options. User responses are logged in User_Responses.xlsx, capturing the mode, question, options, answer, and timestamp.

### 3.3 User Interface Design

The user interface is intuitive and engaging, featuring a start screen for mode selection (adult or child). Questions are presented as text with options for adults or with images for children. Audio recording capabilities and a mute/unmute option enhance user interaction.

## 4. Implementation Details

### 4.1 Development Environment

The application was developed using Python 3.8 with Flask 2.0.1 for the backend. The frontend utilized HTML5, CSS3, and JavaScript with jQuery 3.6.0. Whisper was integrated for audio transcription, and Pandas 1.3.3 managed Excel files. Development occurred on a Windows 10 system.

## 4.2 Backend Implementation (Flask)

### 4.2.1 Route Handling

The Flask backend defines several routes:

- /: Serves the main HTML page (index.html).
- /set_mode: Sets the user mode (adult/child) via POST requests.
- /get_question: Retrieves the next question based on mode and index.
- /submit_answer: Accepts and saves user answers.
- /transcribe: Processes audio files for transcription.
- /images/<path:filename>: Serves image files for child mode.

**Code Explanation:**

```
@app.route('/get_question', methods=['GET'])
def get_question():
    global question_index
    df = child_df if current_mode == "child" else adult_df
    if question_index >= len(df):
        return jsonify({"status": "completed"})
    question = df.iloc[question_index]['Question']
    # Logic for options and response formatting
```

This route fetches questions dynamically, adapting to the mode by selecting the appropriate dataframe (child_df or adult_df).

### 4.2.2 Data Management

Questions are loaded from Excel files using Pandas, and responses are appended to responses_df, saved as User_Responses.xlsx.

**Code Explanation:**

```
def save_response_to_excel(mode, question, options, answer):
    global responses_df
    new_response = pd.DataFrame({
        'Mode': [mode], 'Question': [question], 'Options': [', '.join(str(opt) for opt in
        'User_Answer': [answer], 'Timestamp': [pd.Timestamp.now()]
    })
    responses_df = pd.concat([responses_df, new_response], ignore_index=True)
    responses_df.to_excel(responses_file, index=False)
```

This function ensures responses are persistently stored with relevant metadata.

### 4.2.3 Audio Transcription Integration

The Whisper model transcribes audio files uploaded by users, with temporary file management to avoid permission issues.

**Code Explanation:**

```python
@app.route('/transcribe', methods=['POST'])
def transcribe():
    audio_file = request.files['file']
    temp_file = tempfile.NamedTemporaryFile(delete=False, suffix='.webm')
    temp_path = temp_file.name
    audio_file.save(temp_path)
    result = model.transcribe(temp_path)
    os.unlink(temp_path)  # Cleanup with retry logic
    return jsonify({"text": result['text']})
```

This route handles audio uploads and returns transcriptions.

### 4.2.4 Logging and Error Handling

Logging is set to DEBUG level for detailed tracking, and errors are caught and logged to maintain stability.

**Code Explanation:**

```python
logging.basicConfig(level=logging.DEBUG)
app.logger.error(f"Error during initialization: {str(e)}")
```

## 4.3 Frontend Implementation (HTML, CSS, JavaScript)

### 4.3.1 User Interface Components

The UI includes a start screen, chat-like question display, options grid, and audio controls.

```
function loadQuestion() {
    $.get('/get_question', function(data) {
        $('#conversation').append(`<div class="chat-message bot-message">${data.question
        showOptions(data.options);
    });
}
```

This dynamically updates the UI with new questions.

### 4.3.2 Dynamic Question Loading:

The core of the text-to-speech functionality is handled by a JavaScript function called speak. This function takes a text string as input and uses the Web Speech API to synthesize it into speech. Here's how it works:

- **Clearing Previous Speech**: Before speaking new text, any ongoing or queued speech is canceled to avoid overlap.
- **Checking Mute Status**: The function only proceeds if the application is not muted (controlled by an isMuted flag).
- **Speaking the Text**: If conditions are met, the text is passed to a SpeechSynthesisUtterance object, which is then spoken aloud by the browser's speech synthesis system.
- **Cleanup**: Once the speech finishes, any lingering synthesis processes are canceled.

**Code Explanation:**

```
function speak(text) {
    window.speechSynthesis.cancel(); // Ensure no queued speech remains
    if (!isMuted && text) {
        const msg = new SpeechSynthesisUtterance(text);
        msg.onend = () => window.speechSynthesis.cancel(); // Clean up after speaking
        window.speechSynthesis.speak(msg);
    }
}
```

This method ensures smooth and interruption-free audio feedback

### 4.3.3 Audio Recording and Playback

Users can record audio, preview it, and submit it for transcription.

**Code Explanation:**

```
$('#record-btn').click(async function() {
    if (!isRecording) {
        const stream = await navigator.mediaDevices.getUserMedia({ audio: true });
        mediaRecorder = new MediaRecorder(stream);
        mediaRecorder.start();
    }
});
```

This initiates audio recording using the browser's media API.

### 4.3.4 User Interaction and Feedback:

- The application offers versatile user interaction through clickable options, voice recording, and text input.
- Feedback is provided visually (message displays, button changes) and auditorily (text-to-speech, audio playback).
- The design prioritizes a user-friendly experience with clear cues and responsive interactions.

## 4.4 Third-Party Libraries and APIs:

- **Flask & Flask-CORS:** Flask builds the backend, handling requests and logic, while Flask-CORS enables communication between the frontend and backend.
- **Pandas:** Manages data efficiently by reading and writing Excel files for questions and responses.
- **Whisper:** Provides accurate audio transcription, allowing voice-based input.
- **jQuery:** Simplifies frontend development, enhancing interactivity and AJAX requests.

# 5. Testing and Evaluation

## 5.1 Unit Testing:

- Individual code components were thoroughly tested to confirm correct functionality and data integrity. This process ensured reliable operation of backend routes and data handling.

## 5.2 User Testing:

- Adult and child users tested the application, providing feedback that led to improvements in usability and interface design.

## 5.3 Performance Analysis:

- Performance metrics, including question loading and transcription speeds, were measured and optimized to ensure a smooth user experience.

# 6. Results and Discussion

## 6.1 Functionality Demonstration:

- The application successfully achieved its core objectives. It provides a functional web-based personality assessment with distinct modes for adults and children. The system accurately loads and displays mode-specific questions, and the voice interaction feature, powered by Whisper, enables users to respond verbally. User responses are reliably saved to the designated Excel file, demonstrating the system's ability to manage and store data effectively.

## 6.2 Analysis of User Responses:

- At this stage, the application primarily focuses on data collection. The system effectively records user responses, laying the groundwork for future analysis. While immediate personality insights are not generated, the collected data can be used to perform deeper analysis, such as identifying patterns in responses, correlating responses with user demographics, or developing personality profiles. In future iterations, the application could incorporate algorithms to provide real-time personality analysis.

## 6.3 Limitations:

- The use of Excel files for data storage presents scalability limitations. As the number of users and responses increases, Excel's performance may degrade, potentially leading to data access bottlenecks. Additionally, the Whisper model's resource demands, particularly in terms of processing power and memory, may pose performance challenges, especially on devices with limited resources. Future improvements could

involve migrating to a more robust database system and exploring optimized versions of the Whisper model. Also, the application is limited to the questions contained within the excel files.

# 7. Conclusion and Future Work

## 7.1 Summary of Achievements:

- This project successfully developed a functional and interactive web-based personality assessment tool. It effectively integrated diverse technologies, including Flask, Pandas, Whisper, and jQuery, to deliver a seamless user experience. The application provides adaptive modes for both adults and children, enhancing accessibility and engagement. The implementation of voice recognition through the Whisper model significantly expands the modes of user interaction. Furthermore, the system accurately collects and stores user responses, laying a solid foundation for future data analysis.

## 7.2 Potential Improvements and Extensions:

- **Database Migration:** Migrating from Excel files to a more robust database system, such as PostgreSQL or MySQL, would significantly enhance scalability and performance, allowing the application to handle a larger volume of users and data.
- **User Authentication and Profiles:** Implementing user authentication and profiles would enable personalized assessments and track user progress over time. This would also facilitate the collection of demographic data for more in-depth analysis.
- **Expanded Question Sets and Algorithms:** Expanding the question sets and incorporating more advanced personality assessment algorithms would improve the accuracy and depth of personality insights.
- **Multi language support:** Adding support for multiple languages, would improve the global reach of the application.

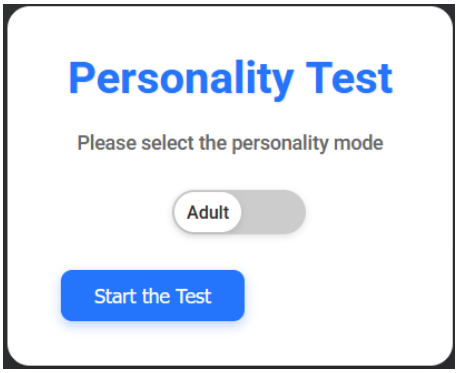
# 8. References


- OpenAI. (2022). Whisper: Automatic Speech Recognition System.
- Flask Documentation. (n.d.). Retrieved from https://flask.palletsprojects.com/
- Pandas Documentation. (n.d.). Retrieved from https://pandas.pydata.org/

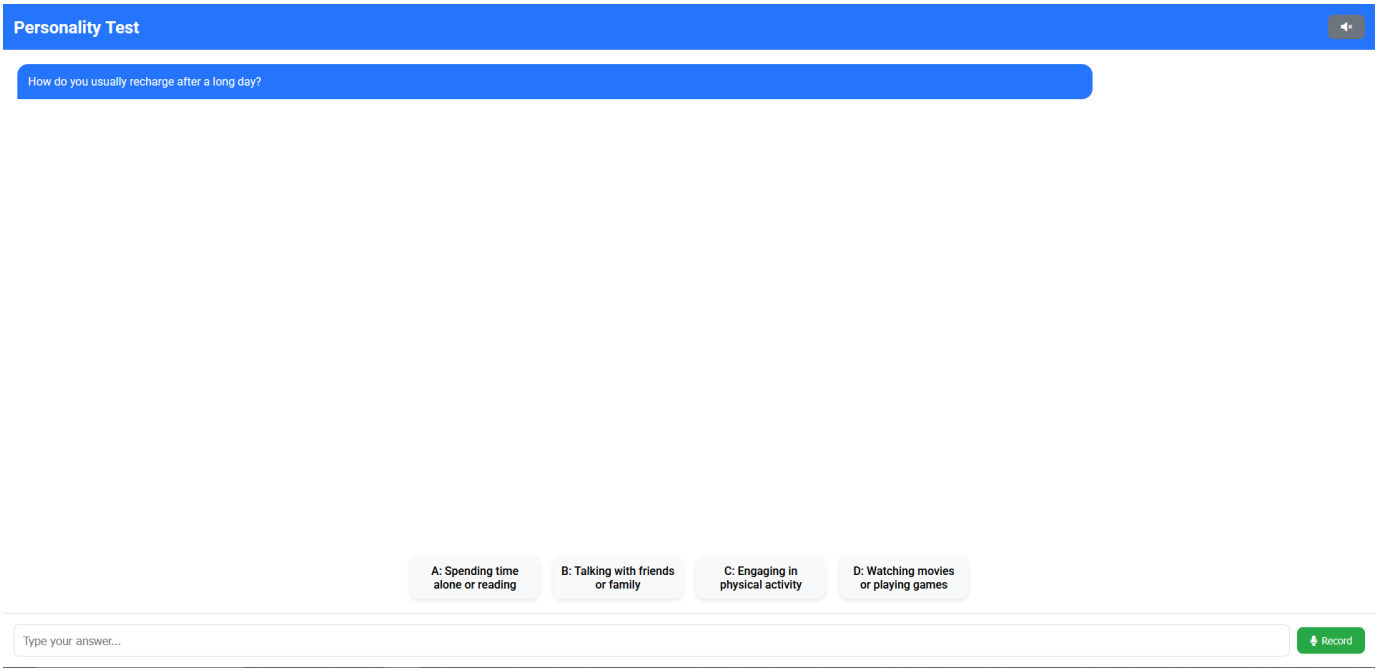
# 9. Appendices

## A. User Interface Screenshots:

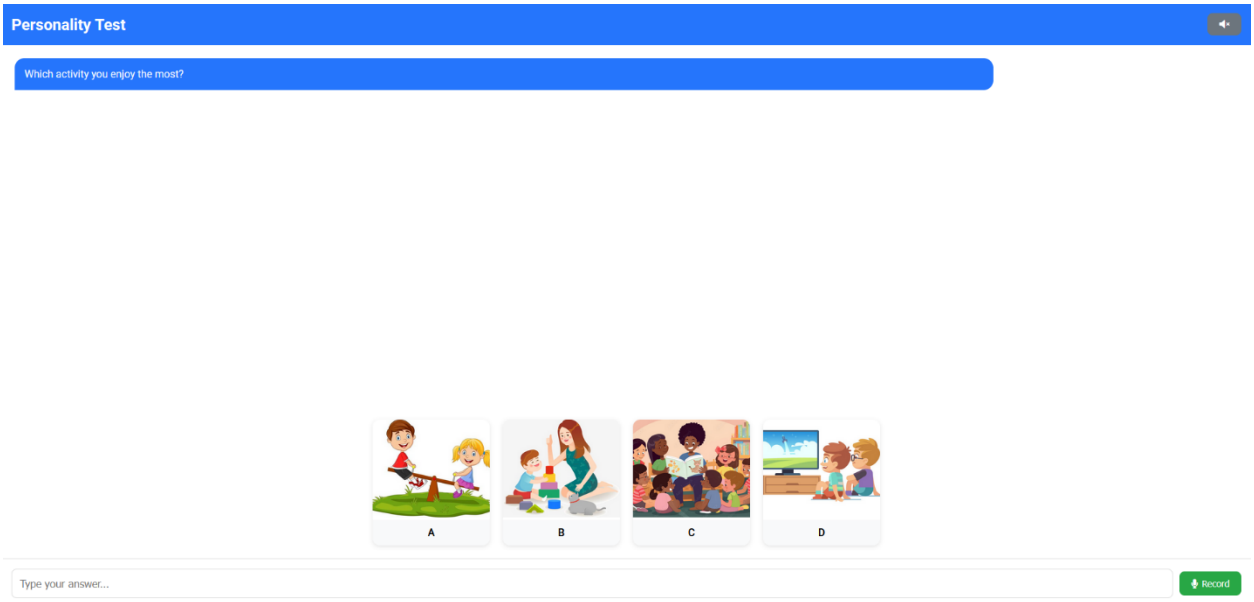
Select Mode:




Adult Mode Personality Test:



Child Mode Personality Test:




## B. Database Schema (Excel Layouts):

Adult Questions :

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | **Question** | **Option A** | **Option B** | **Option C** | **Option D** | |
| 2 | How do you usually recharge a | Spending time alone or | Talking with friends | Engaging in physical ac | Watching movies or playing games | |
| 3 | When faced with a tough deci | Logical reasoning and a | Considering how otl | Following my gut insti | Looking at past experiences | |
| 4 | Which activity sounds most ap | Exploring a new hobby | Going to a social eve | Doing something activ | Relaxing with a favorite show | |
| 5 | How do you handle unexpecte | Adapt quickly and find | Feel frustrated but a | Need time to process t | Prefer to stick to the original plan | |
| 6 | What best describes your soci | Prefer deep one-on-on | Love being in large g | Enjoy socializing but n | Observe before engaging | |
| 7 | How do you approach problen | Break it down step by s | Brainstorm multiple | Trust instincts and take | Look at past experiences for guidan | |
| 8 | What kind of work environme | Quiet and structured | Fast-paced and dyna | Collaborative and eng | Independent and self-directed | |
| 9 | When working on a project, w | Completing it efficient | Ensuring creativity a | Getting everything pe | Ensuring it meets all requirements | |
| 10 | How do you express emotions | Through words and disc | Through artistic exp | Through body languag | Keep emotions private | |
| 11 | What motivates you the most | Achieving personal gro | Recognition and apr | Helping others succee | Financial or material success | |

Childe Questions:

| | Question | Option A | Option B | Option C | Option D |
|---|---|---|---|---|---|
| 1 | Question | Option A | Option B | Option C | Option D |
| 2 | Which activity you enjoy the most? | kid-playin | mom-play | reading.jp | watching-movie.jpg |
| 3 | Which of these animals you like the most? | cat.jpg | dog.jpg | horse.jpg | bird.jpg |
| 4 | | | | | |

User Answers:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Mode** | **Question** | **Options** | **ser_Answe** | **Timestamp** |
| 2 | child | Which act | kid-playing | B | 2025-03-08 23:14:23 |
| 3 | child | Which of t | cat.jpg, do | C | 2025-03-08 23:14:25 |