**A**

**Project Report**

on

**Manobal : A Personality Development App**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

**SESSION 2024-25**

in

**Computer Science and Engineering**

By

Prince Kumar (2102310100075)

Nitish Kumar (2102310100070)

Khushi Chaudhary (2102310100055)

Prabhat Chaudhary (2102310100073)

**Under the guidance of**

Mr. Hemant Kumar Bhardwaj



**R D Engineering College, Ghaziabad**

Affiliated to



**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**(Formerly UPTU)**

**MAY, 2025**

**DECLARATION**

*We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature*

*Name : Prince Kumar*

*Roll No. : 2102310100075*

*Date : 23-may-2025*

*Signature*

*Name : Nitish Kumar*

*Roll No. : 2102310100070*

*Date : 23-may-2025*

*Signature*

*Name : Khsuhi Chaudhary*

*Roll No. : 2102310100055*

*Date : 23-may-2025*

*Signature :*

*Name : Prabhat Chaudhary*

*Roll No. : 2102310100073*

*Date : 23-may-2025*

**CERTIFICATE**

This is to certify that *Project Report entitled –* ***“Manobal”***which is submitted by *Prince Kumar* **(2102310100075), Khushi Chaudhary (2102310100055),Nitish Kumar (2102310100070),Prabhat Chaudhary (2102310100073)**in partial fulfillment of the requirement for the award of degree B. Tech. in Department of CSE & CSE Allied, of Dr. A.P.J. Abdul Kalam Technical University, U.P., Lucknow., is a record of the candidate own work carried out by him/her under my/our supervision. The matter embodied in this Project report is original and has not been submitted for the award of any other degree.

### Name of Guide:

### Mr Hemant Kumar Bhardwaj Prof. Lav Kumar Dixit

### (Associate Prof.) (Head, CSE & CSE Allied)

### Date: 23-May-2025

**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B.Tech Final Year. We owe special debt of gratitude to our guide*

***Mr. Hemant Kumar Bhardwaj****, Department of CSE, R.D. Engineering College, Ghaziabad for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen light of the day.*

*We express our sincere gratitude to* ***Prof. Lav Kumar Dixit****, HoD, Department of CSE & CSE Allied, R.D. Engineering College, Ghaziabad, for his stimulating guidance, continuous encouragement and supervision during the development of the project.*

*We are extremely thankful to* ***Prof. Mohd. Vakil****, Dean Academics, R.D. Engineering College, Ghaziabad, for his full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

|  |  |  |
| --- | --- | --- |
| *Signature*  *Name* | *:*  *: Prince Kumar* | *Signature :*  *Name : Nitish Kumar* |
| *Roll No.* | *: 2102310100075* | *Roll No. : 2102310100070* |
| *Date* | *:* 23-May-2025 | *Date :* 23-May-2025 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Signature*  *Name* | *:*  *: Khushi Chaudhary* | *Signature*  *Name* | *:*  *: Prabhat Chaudhary* |
| *Roll No.* | *: 2102310100055* | *Roll No.* | *: 2102310100073* |
| *Date* | *:* 23-May-2025 | *Date* | *:* 23-May-2025 |

## ABSTRACT

In today's fast-paced digital world, personality development is increasingly recognized as a critical component for personal and professional success. Traditional methods for enhancing soft skills often lack personalization, engagement, and practical applicability.

To address these challenges, we have developed **Manobal**, a mobile application designed to assess and enhance user personalities using the scientifically validated **OCEAN model**, also known as the Big Five Personality Traits.

The OCEAN model evaluates users across five key dimensions: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Based on an initial dynamic assessment, Manobal generates customized learning tasks tailored to the user's dominant personality traits. The system also includes community interaction features to promote peer learning and user engagement.

Built using **React Native** for cross-platform development and **Node.js** with **MongoDB** for backend services, Manobal integrates secure authentication via JWT and uses structured logic to assign tasks without relying on complex AI models. Ethical considerations like data privacy, transparency, and user consent are prioritized in the app’s design.

This report details the conceptual foundation, existing systems, methodology, implementation, results, and future directions of Manobal. The app not only provides actionable insights for users but also serves as a deployable solution for institutions seeking structured personality development tools.

## TABLE OF CONTENT

**Page No**

**DECLARATION ii**

### CERTIFICATE iii

### ACKNOWLEDGEMENT iv

### ABSTRACT v

### LIST OF FIGURES viii

### LIST OF TABLES ix

**LIST OF ABBREVIATIONS x**

### CHAPTER 1 INTRODUCTION 1

* 1. Introduction to soft skills and their importance 1
  2. Role of AI in Skill Development 2
  3. Rise of Persoanlised Apps 3
  4. Introduction to OCEAN Model 4
  5. Problem Statement: Lack of adaptive, engaging, and 5

peer-interactive platforms

1.6 Objective of the “Manobal” App

### CHAPTER 2 LITERATURE REVIEW 6

1. Review of existing app s like “Bestify Me”, “Make me Better” 6
2. Limitations of traditional learning systems 7
3. Detailed analysis of OCEAN-based tools 8
4. Problem Defintion and Scope 9

### CHAPTER 3 PROPOSED METHODOLOGY 10

* 1. Module-wise structure 10
     1. Authentication Module 11
     2. Assessment Module (Dynamic OCEAN) 12
     3. Task Generation Module 13
     4. Dashboard and Feedback 14
     5. Community Interaction 15
  2. Diagrams 16
     1. ER Diagram 17
     2. DFD (Level 0 &1) 18
     3. Use Case Diagram 19
     4. Activity Diagram 20
     5. Flowcharts 21
  3. Tools and Tect Used: React Native, Node.js, MongoDB, JWT, BcryptJS 22
  4. Dataset Description 23
  5. Security Connsiderations 24
  6. Feasibility Study (Technical, Operational, Economic) 25

### CHAPTER 4 IMPLEMENTATION AND RESULTS 26

4.1 Screenshots of: 26

4.1.1 Login/Register 27

4.1.2 Assessnebt Interface 28

4.1.3 Dashboard 29

4.1.4 Task Assignment 30

4.1.5 Community Posts 31

4.1.6 Feedback page 32

4.2 Implementation details (how each module coded and works) 33

4.3 Results Analysis 34

4.3.1 User onboarding stats 35

4.3.2 Assessment effectiveness 36

4.3.3 Community interaction rate 37

### CHAPTER 5 IMPLEMENTATION AND RESULTS 38

5.1 Summary of what was achieved 39

5.2 Key features of Manobal App 40

5.3 Limitations 41

5.4 Future Enhacements 42

5.4.1 Integration of voice recognition 43

5.4.2 AI-based task adaption 44

5.4.3 Real-time feedback systems 45

5.4.4 Support for multiple languages 48

### REFERENCES 50

### APPENDIX A SCREEN SHOT, CODESSION 51

### APPENDIX B RESEARCH PAPER 52

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| Figure 1.1 | Name of Figure | 22 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No**. |
| 1.1 | Name of Table | 22 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ABBREVIATIONS**

**AI** Artificial Intelligence

**UI** User Interface

**ERP** Enterprise resource planning

**CRM** Customer relationship management

**CHAPTER 1**

**Introduction**

* 1. **Introduction to Learning**

In the modern world, where technical expertise is abundant, soft skills have emerged as a vital differentiator in both personal and professional domains. While hard skills refer to the technical knowledge and abilities required for specific tasks, soft skills encompass interpersonal qualities, communication abilities, emotional intelligence, adaptability, and other traits that define how individuals interact with others.

**Soft skills** play a crucial role in determining career success, leadership potential, teamwork effectiveness, and overall workplace harmony. Skills such as effective communication, problem-solving, critical thinking, empathy, and time management are highly valued across industries. Employers increasingly seek candidates who not only have technical qualifications but also exhibit strong interpersonal capabilities to navigate complex, collaborative environments.

The importance of soft skills extends beyond professional settings. In personal life, these skills influence relationships, decision-making, conflict resolution, and emotional well-being. Individuals with well-developed soft skills are better equipped to manage stress, adapt to change, and contribute positively to their communities.

Despite their significance, traditional educational systems often focus heavily on technical and academic content, leaving little room for structured personality development. This gap necessitates the integration of tools and technologies that can nurture soft skills in a personalized and engaging manner.

In response to this need, **Manobal** has been developed as a comprehensive personality development platform. By leveraging the **OCEAN model** of personality traits, it assesses users and guides them through curated tasks designed to strengthen their unique personality profile. Through this approach, Manobal aims to make soft skills development more accessible, practical, and impactful in the digital age.

* 1. **Role of AI in Skill Development**

Artificial Intelligence (AI) is revolutionizing nearly every aspect of human life, and skill development is no exception. Traditional methods of learning and self-improvement often adopt a one-size-fits-all approach, lacking personalization, real-time feedback, and adaptability. AI bridges this gap by introducing intelligent, data-driven systems that adapt to individual learning styles, pace, and personality traits.

In the context of soft skills and personal growth, AI enables:

* Personalized Learning Paths: AI can analyze user behavior, preferences, and progress to tailor content and tasks that align with the user’s needs and goals.
* Real-Time Feedback: Through voice recognition, facial expression analysis, and natural language processing (NLP), AI systems can provide immediate, meaningful feedback to users on communication skills, emotional tone, and more.
* Predictive Analytics: By assessing patterns in responses and behavior, AI can predict areas of strength and weakness, guiding users to focus on relevant skill sets.
* Gamification and Engagement: AI-driven interfaces can enhance engagement through interactive modules, adaptive difficulty levels, and reward-based learning experiences.

Moreover, AI fosters continuous learning by adapting to changes in user behavior over time, making it suitable for long-term development. In applications like Manobal, AI’s potential is evident through dynamic assessments based on the OCEAN model, which help in crafting task recommendations tailored to a user's personality traits.

The integration of AI into skill development not only increases efficiency and personalization but also democratizes access by enabling self-paced, self-directed learning from anywhere. As AI continues to evolve, it promises to make human development more intuitive, scalable, and impactful than ever before.

* 1. **Rise of Personalised Apps**

In recent years, there has been a significant shift from generic digital platforms to personalized applications that tailor content and experiences based on individual preferences, behaviors, and goals. This transformation is driven by the increasing demand for customized solutions that enhance engagement, learning, and productivity.

Personalized apps utilize data analytics, user feedback, and machine learning algorithms to adapt in real-time to user needs. Whether it's in education, fitness, mental health, or personal development, users expect platforms that understand and evolve with them. Apps like Duolingo, Headspace, and MyFitnessPal have set benchmarks in delivering highly tailored user experiences.

In the domain of personality development, personalization is especially valuable. Every individual has unique traits, learning styles, and growth areas. A one-size-fits-all model often fails to inspire meaningful progress. Manobal addresses this gap by using personality assessments based on the OCEAN model to recommend targeted tasks and learning paths. This ensures that each user receives content aligned with their personality, making the journey of self-improvement both effective and engaging.

The rise of personalized apps marks a new era of user-centric design—where technology empowers individuals to grow in ways that are most relevant to them.

* 1. **Introduction to OCEAN Model**

The OCEAN model, also known as the Big Five Personality Traits, is one of the most widely accepted and scientifically validated frameworks for understanding human personality. It categorizes personality into five key dimensions that reflect different aspects of an individual's character and behavior:

1. **Openness to Experience** – Reflects creativity, curiosity, imagination, and a preference for novelty and variety.
2. **Conscientiousness** – Indicates a high level of self-discipline, organization, and a goal-oriented approach.
3. **Extraversion** – Represents sociability, energy, and the tendency to seek stimulation in the company of others.
4. **Agreeableness** – Involves being compassionate, cooperative, and friendly rather than suspicious and antagonistic.
5. **Neuroticism** – Describes emotional instability, anxiety, moodiness, and a tendency to experience negative emotions.

Each trait is measured on a continuum, meaning individuals may score high, low, or somewhere in between for each trait, allowing for a detailed and balanced personality profile.

The strength of the OCEAN model lies in its universality and reliability. It has been validated across different cultures and has applications in psychology, education, human resources, and technology. In recent years, it has also been integrated into digital applications for personalized learning, mental health support, and behavior prediction.

In Manobal, the OCEAN model is used as the foundation for assessing user personalities. Based on their assessment results, users are assigned customized development tasks aligned with their dominant traits. This targeted approach makes the learning process more effective, engaging, and relevant to the user's personal growth goals.

* 1. **Problem Statement: Lack of adaptive, engaging, and peer-interactive platforms**

In today’s competitive and fast-paced environment, soft skills such as communication, leadership, emotional intelligence, and adaptability have become essential for personal and professional success. However, most traditional methods of personality development rely on generic training modules, static personality tests, and manual evaluation systems. These approaches often lack personalization, fail to maintain user engagement, and ignore the importance of peer interaction, which is vital for practical and long-lasting behavioral change.

Existing mobile applications in the self-improvement or personal development space—like *Bestify Me*, *Make Me Better*, and others—focus mainly on passive content delivery such as daily articles or motivational quotes. While informative, these platforms do not offer customized learning paths, adaptive feedback mechanisms, or task-based reinforcement aligned with individual personality profiles. Moreover, they rarely facilitate community-driven learning or interactive peer support systems, both of which are key to building confidence, sharing perspectives, and learning through collaboration.

Additionally, current tools do not effectively utilize established psychological models such as the OCEAN personality framework, nor do they integrate modern technologies like AI or structured logic to enhance decision-making, user tracking, and growth measurement. This results in a disconnect between assessment and actionable outcomes, causing users to disengage due to lack of structure or progress.

Therefore, there is a pressing need for a comprehensive, adaptive, and interactive platform that not only assesses personality traits accurately but also recommends actionable tasks, promotes community participation, and respects user privacy and ethics.

The proposed solution, Manobal, is a mobile application designed to bridge this gap. By leveraging the OCEAN model, structured logic, and community features, Manobal aims to provide a highly personalized, engaging, and socially interactive platform for personality development, especially targeting students, professionals, and individuals seeking structured self-growth.

* 1. **Objective of “Manobal” App**

The rapid advancement of technology, coupled with the increasing demand for soft skills in personal and professional life, highlights the urgent need for intelligent platforms that go beyond traditional learning. Manobal is envisioned as a holistic personality development mobile application that bridges the gap between psychological theory and real-world application. The app leverages the scientifically proven OCEAN model (Big Five Personality Traits) to deliver a highly personalized, task-oriented, and community-supported self-improvement experience.

The detailed objectives of the Manobal application are as follows:

1. To provide accurate and structured personality assessment using the OCEAN model

The foundation of Manobal lies in understanding each user’s personality through the five core dimensions of the OCEAN model:

* Openness: Creativity and willingness to try new things
* Conscientiousness: Discipline, organization, and goal-directed behavior
* Extraversion: Social interaction and assertiveness
* Agreeableness: Empathy, cooperation, and compassion
* Neuroticism: Emotional regulation and resilience

By deploying a carefully designed assessment based on the Big Five Inventory, the app evaluates user traits and establishes a baseline for development.

2. To dynamically generate customized tasks that align with individual personality traits

After identifying dominant personality traits, Manobal assigns relevant tasks from a pre-configured bank tailored to the user's strengths and areas of improvement. For instance:

* A user high in Openness might receive tasks like creative writing or idea generation.
* A user high in Conscientiousness might receive schedule planning or habit-building exercises.
* Someone scoring high on Extraversion might be guided to participate in public speaking or group activities.

This functionality ensures every user experiences a development path that feels uniquely relevant and motivating.

3. To maintain consistent user engagement through adaptive learning mechanisms

Static learning content often leads to disengagement. Manobal addresses this by implementing an adaptive learning structure, where task complexity and focus areas evolve as the user progresses. A feedback loop collects data on task completion and user satisfaction, allowing future recommendations to be better aligned with growth trends.

4. To foster a sense of community through collaborative features

Self-improvement becomes more effective when reinforced by social support. Manobal integrates a community module where users can share insights, write short blogs, discuss challenges, and support each other. This peer-to-peer interaction enhances accountability, confidence, and learning through shared experience.

5. To build a transparent and ethical assessment system

Many AI tools rely on opaque “black-box” algorithms that offer little understanding to users about how decisions are made. Manobal ensures complete transparency by using a rule-based, interpretable scoring system instead of hidden AI models. Users understand exactly how their responses lead to certain results and recommendations, promoting trust and user empowerment.

6. To provide a lightweight, accessible solution for diverse user environments

Manobal is designed to run on low-resource devices and can operate with minimal or no internet connectivity. This makes it suitable for students and institutions in rural or semi-urban settings, where access to high-end infrastructure may be limited.

7. To enable long-term personal growth through continuous feedback and progress tracking

Growth is not a one-time event but a continuous process. Manobal supports ongoing personality development by maintaining logs of completed tasks, offering progress dashboards, and enabling users to reflect on their growth over time. This persistent model encourages habit formation and self-reflection.

By achieving these objectives, Manobal not only differentiates itself from existing self-improvement tools but also serves as a replicable model for ethical, accessible, and effective personality development solutions.

## 

## CHAPTER 2

**Exiting System/Literature Review**

**Hint:** **Student will write here about Existing system. Existing system means you have knowledge about existing system and you are going to propose new system. It is also called literature review. This chapter will show that you are aware about existing system or not.**

**After review of existing system students will writea paragraph about challenges and scope with exixting system. In last paragraph of this chapter you will mention problem definition which you are going to do.**

**Number of pages-** 4 to 15 pages

## CHAPTER 3

**Proposed Methodology**

**Hint: This chapter is very important for you. This chapter describe your proposed methodology that is used in the project in the form of various descriptions depending upon project taken.your project may be some web application, mobile appliocation, technology based, algorithm based, some new research etc.**

**You have to cover the points from given below, depend upon your project**

1. **Module description**
2. **Each field descrioton of each module/submodule**
3. **ER diagram**
4. **DFD**
5. **Flow chart**
6. **Algorithm**
7. **Equation**
8. **Formula**
9. **Datasets description**
10. **Use case diagram**
11. **Activity diagram**
12. **etc**

**--------------------**

1. **Feasiblity study**
2. **H/W and S/W used**

## CHAPTER 4

## Implementation and Results

**Hint: In this chapter students can put (i) screen shots with descrition if your project is web application or mobile application (ii) results and result analysis if your project is technology/algorithm/research based.**

## CHAPTER 5

**Conclusion and Future Scope**

**Hint: Here, students will conclude the project report in one or two paragraphs**

**References**

[1] Goldberg, L. R. (1993). Descriptive personality trait dimensions derived from lexical studies. A foundational exploration of the OCEAN framework.

[2] McCrae, R. R., & Costa, P. T. (1997). Universal features of personality traits across cultures. Demonstrates stability of Big Five across global populations.

[3] Journal study on linguistic expression and psychological attributes among long-form writers. Offers insights on personality-word correlations in textual content.

[4] Investigative report on the predictive power of digital behaviour patterns on private traits. Highlights privacy concerns in data-driven models.

[5] Study on social media language and its correlation to Big Five traits, introducing automated evaluation tools.

[6] Comparative evaluation of personality detection models applied to social media behaviour logs. Suggests feature limitations in algorithmic inference.

[7] Analysis of profile image preferences and trait signals, adding a visual dimension to digital personality profiling.

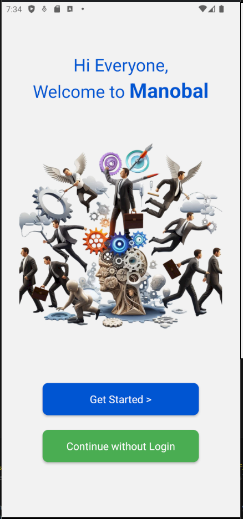
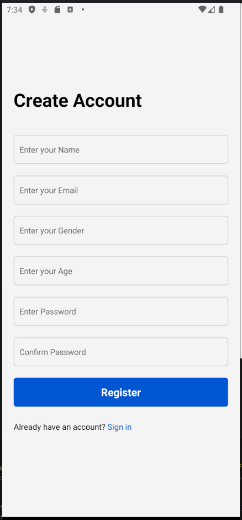
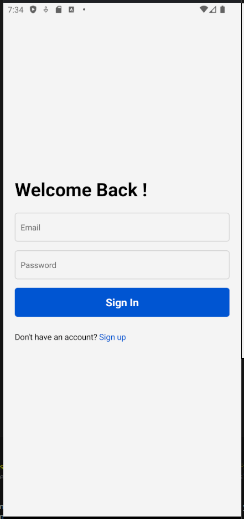
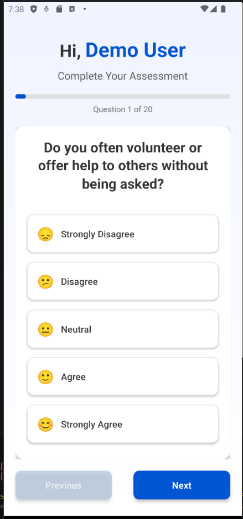
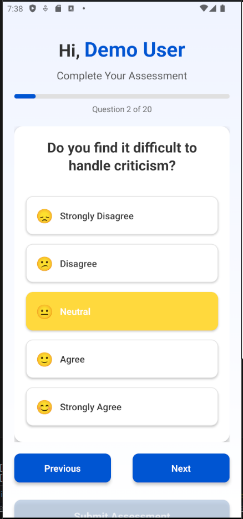
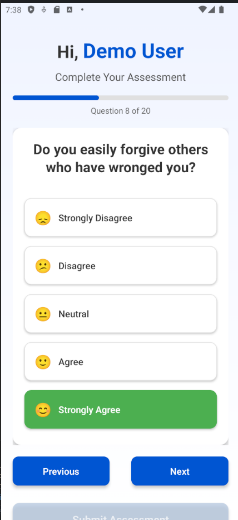
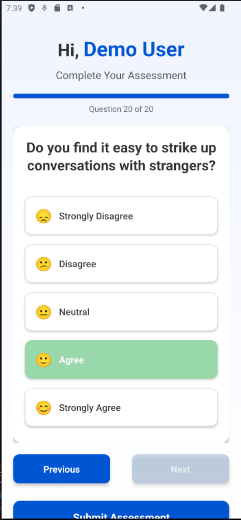
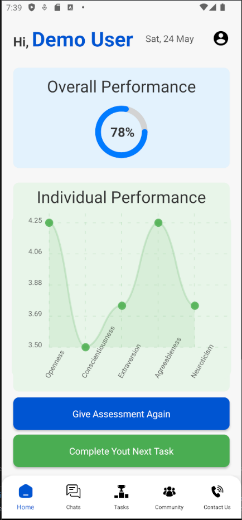
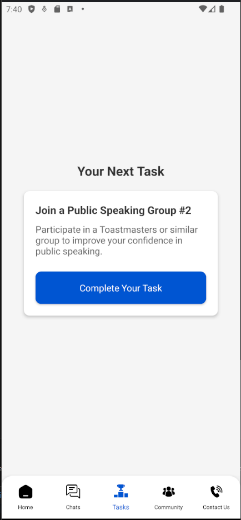
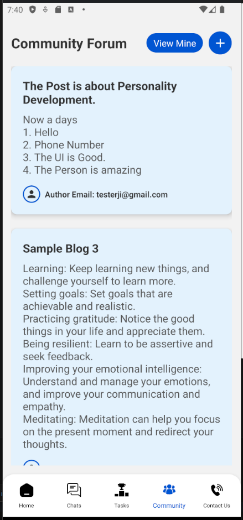
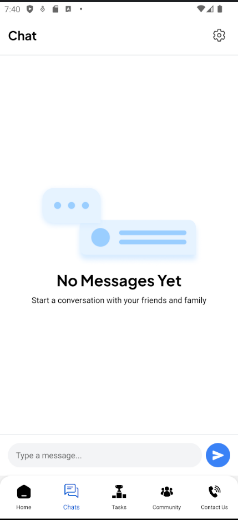
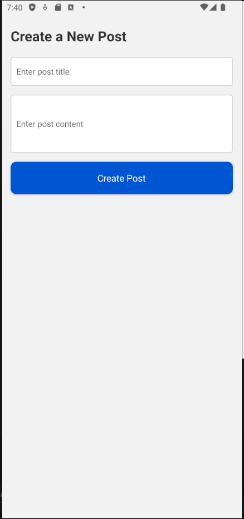
[8] Research on hybrid methods for predicting traits using text processing and deep neural networks. Demonstrates strengths and drawbacks of black-box models.

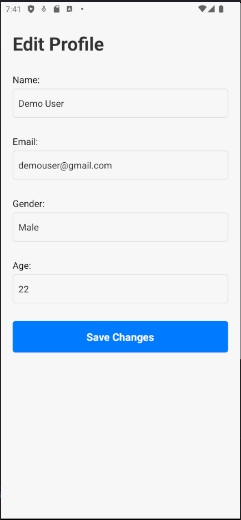
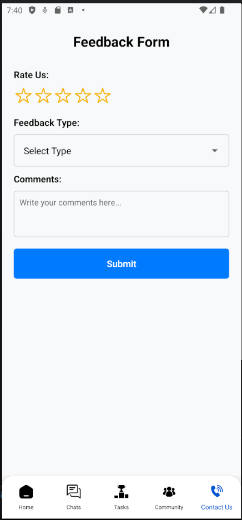
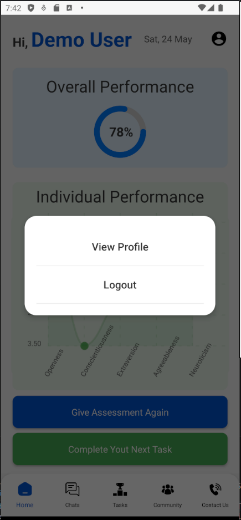
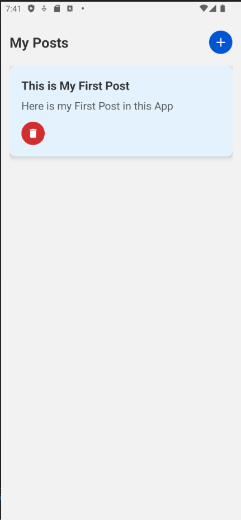
[9] Document modelling strategies for personality classification through deep learning techniques. Explores document-level semantics in psychological detection.

[10] Evaluation of personality detection using combined visual, audio, and text modalities. Emphasizes the role of multimodal AI in trait classification.

**Appendix-A**

* 1. **Screenshots of the App**

**           **

**   **

* 1. **Source Code of App**

**Server.js***// Import*

*const* express = require('express');

*const* dotenv = require('dotenv');

*const* cors = require('cors');

*const* mongoose = require('mongoose');

*const* authRouter = require('./routes/authRoute');

*const* mainRouter = require('./routes/mainRoute');

*// Important Calls*

*const* app = express();

dotenv.config();

app.use(cors());

app.use(express.json());

*// Declarations*

*const* PORT = process.env.PORT || 8000;

*const* HOST = process.env.HOST;

*const* MONGO\_URL = process.env.MONGO\_URL;

*// API Routes*

app.use('/api/auth', authRouter);

app.use('/api/main', mainRouter);

*// Server Listen & DB Connection*

mongoose

  .connect(MONGO\_URL)

  .then(() => {

    console.log('DB is Connected');

    app.listen(PORT, () => {

      console.log(`Server is Running on http://${HOST}:${PORT}`);

    });

  })

  .catch((error) => {

    console.log(error);

  });

**authRoute.js**  
*const* express = require('express');

*const* { register, login } = require('../controllers/authController');

*const* authRouter = express.Router();

authRouter.post('/register', register);

authRouter.post('/login', login);

module.exports = authRouter;

**mainRoute.js**

*const* express = require('express');

*const* mainRouter = express.Router();

*const* {

  assessment,

  questions,

  user,

  score,

  posts,

  myposts,

  createpost,

  deletepost,

  getnexttask,

  completeTask

} = require('../controllers/mainController');

*const* authMiddleware = require('../middlewares/authMiddleware');

mainRouter.post('/assessment', authMiddleware, assessment);

mainRouter.get('/get-questions', questions);

mainRouter.get('/get-user', authMiddleware, user);

mainRouter.get('/get-score', authMiddleware, score);

mainRouter.get('/fetch-posts', authMiddleware, posts);

mainRouter.get('/fetch-my-posts', authMiddleware, myposts);

mainRouter.post('/create-post', authMiddleware, createpost);

mainRouter.delete('/delete-post/:postId', authMiddleware, deletepost);

mainRouter.get('/get-next-task', authMiddleware, getnexttask);

mainRouter.get('/complete-task', authMiddleware, completeTask);

module.exports = mainRouter;

**authMiddleware.js**

*const* jwt = require('jsonwebtoken');

*const* authMiddleware = (req, res, next) => {

*const* token = req.header('manobal'); *// Bearer <token>*

*if* (!token) {

*return* res.status(401).json({ message: 'No token provided, access denied' });

    }

*try* {

*const* decoded = jwt.verify(token, process.env.JWT\_SECRET); *// Replace with your secret key*

        req.user = decoded; *// Attach user info to the request object*

        next(); *// Pass control to the next middleware or route handler*

    } *catch* (err) {

*return* res.status(401).json({ message: 'Invalid token' });

    }

}; module.exports = authMiddleware;

**authController.js**

*// Imports*

*const* userModel = require('../models/userModel');

*const* bcrypt = require('bcryptjs');

*const* jwt = require('jsonwebtoken');

*// Register API*

*const* register = *async* (req, res) => {

*try* {

*const* { name, email, age, gender, password } = req.body;

*if* (!name || !email || !age || !gender || !password) {

*return* res.status(400).send({

        sucess: false,

        message: 'All fields must me filled',

      });

    }

*const* isUserExist = *await* userModel.findOne({ email });

*if* (isUserExist) {

*return* res.status(400).send({

        sucess: false,

        message: 'This user is already Registered!!',

      });

    }

*const* hashedPassword = *await* bcrypt.hash(password, 10);

*const* user = *await* userModel({

      name,

      email,

      age,

      gender,

      isAssesmentDone:0,

      password: hashedPassword,

    });

*await* user.save();

*return* res.status(200).send({

      sucess: true,

      message: 'User has been registered sucessfully!!',

    });

  } *catch* (error) {

*return* res.status(200).send({

      sucess: false,

      message: error

    });

  }

};

*// Login API*

*const* login = *async* (req, res) => {

*try* {

*const* { email, password } = req.body;

*// Check if fields are filled*

*if* (!email || !password) {

*return* res.status(400).send({

        success: false,

        message: 'All fields must be filled',

      });

    }

*// Find user by email*

*const* isUserExist = *await* userModel.findOne({ email });

*if* (!isUserExist) {

*return* res.status(400).send({

        success: false,

        message: 'This user is not registered!',

      });

    }

*// Check if password matches*

*const* isUserMatched = *await* bcrypt.compare(password, isUserExist.password);

*if* (!isUserMatched) {

*return* res.status(400).send({

        success: false,

        message: 'Invalid credentials',

      });

    }

*// Generate JWT token*

*const* token = jwt.sign(

      { id: isUserExist.\_id, email: isUserExist.email },

      process.env.JWT\_SECRET, *// Use a secret key from environment variables*

      { expiresIn: '1d' } *// Token expiry time*

    );

*return* res.status(200).send({

      success: true,

      message: 'Login successful!',

      token, *// Send the token to the client*

      user: {

        id: isUserExist.\_id,

        name: isUserExist.name,

        email: isUserExist.email,

        isAssesmentDone:isUserExist.isAssesmentDone

      }, *// Optional user data*

    });

  } *catch* (error) {

*return* res.status(200).send({

      success: false,

      message: error

    });

  }

};

module.exports = { register, login };

**mainController.js**

*const* jwt = require('jsonwebtoken');

*const* questionModel = require('../models/questionModel');

*const* scoreModel = require('../models/scoreModel');

*const* userModel = require('../models/userModel');

*const* postModel = require('../models/postModel');

*const* prevTaskModel = require('../models/prevtasksModel');

*const* userTasksModel = require('../models/userTasksModel')

*const* TaskModel = require('../models/tasksModel');

*const* assessment = *async* (req, res) => {

*try* {

*const* { answers } = req.body;

*if* (!answers || !Array.isArray(answers)) {

*return* res.status(400).send({

        success: false,

        message: 'Invalid input: answers must be an array.',

      });

    }

*const* traits = {

      Openness: [],

      Conscientiousness: [],

      Extraversion: [],

      Agreeableness: [],

      Neuroticism: [],

    };

*for* (*const* answer of answers) {

*const* { id, \_id, answer: score } = answer;

*if* (score < 1 || score > 5) {

*return* res.status(400).send({

          success: false,

          message: `Invalid score for question ID ${id}: must be between 1 and 5.`,

        });

      }

*const* ques = *await* questionModel.findOne({ id });

*const* trait = ques.trait;

*if* (trait) {

        traits[trait].push(score);

      }

    }

*const* result = {};

*for* (*const* trait in traits) {

*const* scores = traits[trait];

*let* sum = 0;

*for* (*const* score of scores) {

        sum += score;

      }

*const* average = sum / scores.length;

      result[trait] = average.toFixed(2);

    }

*const* userEmail = req.user.email;

*const* scoreExist = *await* scoreModel.findOne({ userEmail });

*if* (scoreExist) {

*await* scoreModel.deleteOne({ userEmail });

    }

*const* userScore = new scoreModel({ userEmail, scores: result });

*await* userScore.save();

*await* userModel.updateOne({ email: userEmail }, { isAssesmentDone: true });

*return* res.status(200).send({

      success: true,

      message: 'Personality assessment completed successfully!',

      data: result,

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* questions = *async* (req, res) => {

*try* {

*const* ques = *await* questionModel.find();

*const* quesLen = ques.length;

*return* res.status(200).send({

      success: true,

      message: 'Successful!',

      noOfQuestions: quesLen,

      questions: ques,

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* user = *async* (req, res) => {

*try* {

*const* userEmail = req.user.email;

*const* user = *await* userModel.findOne({ email: userEmail });

*if* (!user) {

*return* res.status(400).send({

        success: false,

        message: 'This user is not registered!',

      });

    }

*return* res.status(200).send({

      success: true,

      message: 'User fetched sucessfuly!',

      user: user,

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* score = *async* (req, res) => {

*try* {

*const* userEmail = req.user.email;

*const* userScore = *await* scoreModel.findOne({ userEmail });

*if* (!userScore) {

*return* res.status(400).send({

        success: false,

        message: 'Score is not calculated yet!',

      });

    }

*return* res.status(200).send({

      success: true,

      message: 'Score fetched sucessfuly!',

      score: userScore,

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* posts = *async* (req, res) => {

*try* {

*const* posts = *await* postModel.find().sort({ createdAt: -1 });

*return* res.status(200).send({

      success: true,

      message: 'posts fetched sucessfuly!',

      posts: posts,

    });

  } *catch* (err) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* myposts = *async* (req, res) => {

*try* {

*const* posts = *await* postModel

      .find({ author: req.user.email })

      .sort({ createdAt: -1 });

*return* res.status(200).send({

      success: true,

      message: 'posts fetched sucessfuly!',

      posts: posts,

    });

  } *catch* (err) {

*return* res.status(400).send({

      success: false,

      message: error.message,

    });

  }

};

*const* createpost = *async* (req, res) => {

*try* {

*if* (!req.body.title || !req.body.content) {

*return* res.status(400).json({ error: 'Title and content are required' });

    }

*const* post = *await* postModel({

      title: req.body.title,

      content: req.body.content,

      author: req.user.email,

    });

    post.save();

*return* res.status(200).send({

      success: true,

      message: 'Post created sucessfuly!',

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* deletepost = *async* (req, res) => {

*const* { postId } = req.params;

*try* {

*const* deletedPost = *await* postModel.findByIdAndDelete(postId);

*if* (!deletedPost) {

*return* res.status(400).send({

        success: false,

        message: 'This post is not registered!',

      });

    }

*return* res.status(200).send({

      success: true,

      message: 'Post deleted sucessfuly!',

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* getnexttask = *async* (req, res) => {

*try* {

*const* orderList = ['openness', 'conscientiousness', 'extraversion', 'agreeableness', 'neuroticism'];

*const* userEmail = req.user.email;

*const* existingPendingTask = *await* userTasksModel.findOne({ userEmail, status: 'pending' }).populate('taskId');

*if* (existingPendingTask) {

*return* res.status(200).send({

        success: true,

        message: 'You have an ongoing task!',

        task: existingPendingTask.taskId,

      });

    }

*const* userScore = *await* scoreModel.findOne({ userEmail });

*if* (!userScore) {

*return* res.status(400).send({

        success: false,

        message: 'Score is not registered!',

      });

    }

*const* scores = userScore.scores;

*const* sumScore = Object.values(scores).reduce((acc, score) => acc + score, 0);

*let* userLevel = '';

*if* (sumScore >= 1 && sumScore <= 12) {

      userLevel = 'beginner';

    } *else* *if* (sumScore >= 13 && sumScore <= 18) {

      userLevel = 'intermediate';

    } *else* *if* (sumScore >= 19 && sumScore <= 20) {

      userLevel = 'advanced';

    }

*const* choosenTrait = orderList[Math.floor(Math.random() \* orderList.length)];

*const* prevCompletedTasks = *await* userTasksModel.find({ userEmail, status: 'completed' }).distinct('taskId');

*const* newTask = *await* TaskModel.findOne({

      \_id: { $nin: prevCompletedTasks },

      level: userLevel,

      trait: choosenTrait,

    });

*if* (!newTask) {

*return* res.status(404).send({

        success: false,

        message: 'No suitable task found for the user!',

      });

    }

*const* assignedTask = new userTasksModel({ userEmail, taskId: newTask.\_id });

*await* assignedTask.save();

*return* res.status(200).send({

      success: true,

      message: 'Task successfully fetched!',

      task: newTask,

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

*const* completeTask = *async* (req, res) => {

*try* {

*const* userEmail = req.user.email;

*const* task = *await* userTasksModel.findOne({ userEmail, status: 'pending' });

*if* (!task) {

*return* res.status(400).send({

        success: false,

        message: 'No pending task found for this user!',

      });

    }

    task.status = 'completed';

*await* task.save();

*return* res.status(200).send({

      success: true,

      message: 'Task marked as completed!',

    });

  } *catch* (error) {

*return* res.status(500).send({

      success: false,

      message: error.message,

    });

  }

};

module.exports = {

  assessment,

  questions,

  user,

  score,

  posts,

  myposts,

  createpost,

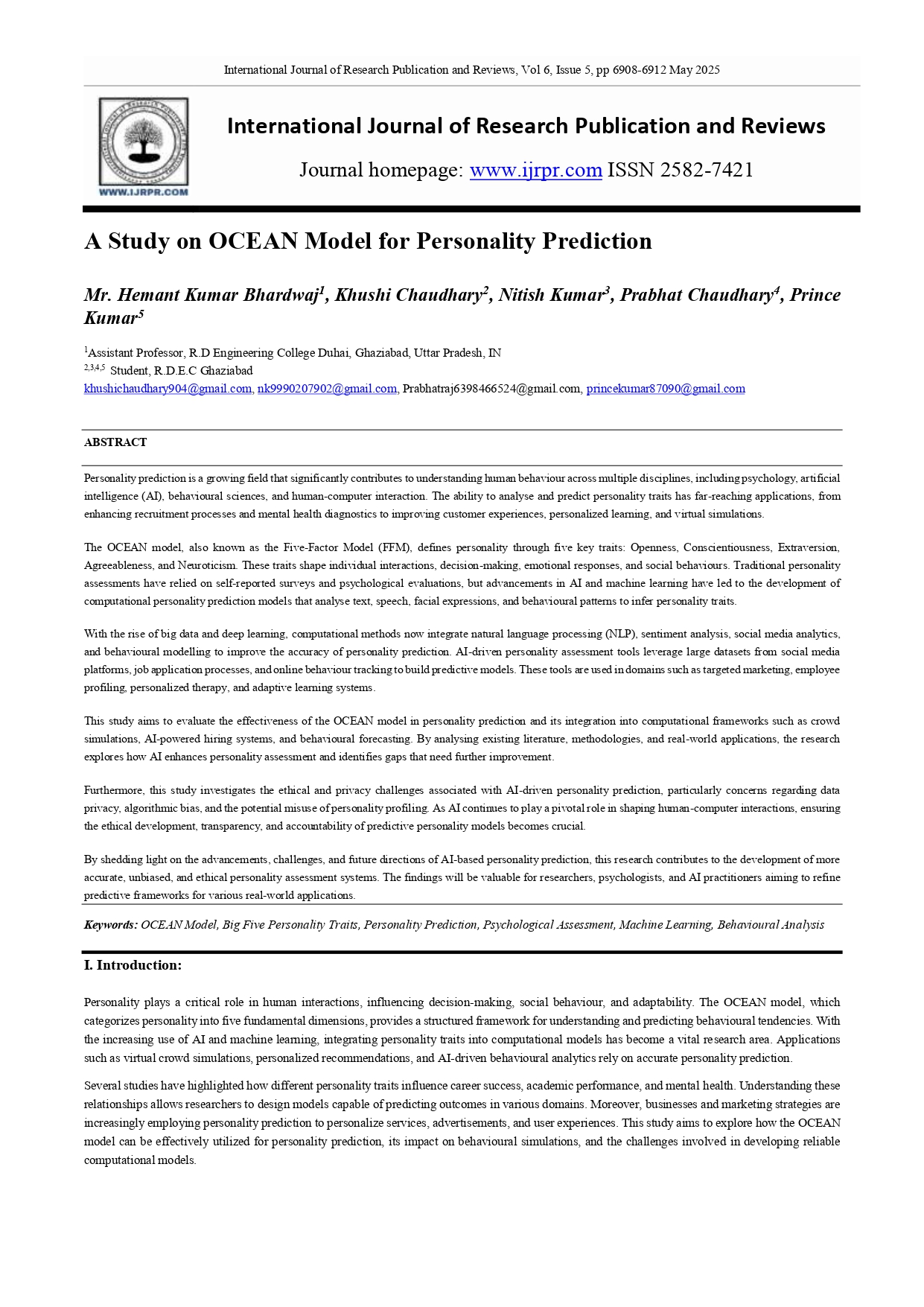
  deletepost,

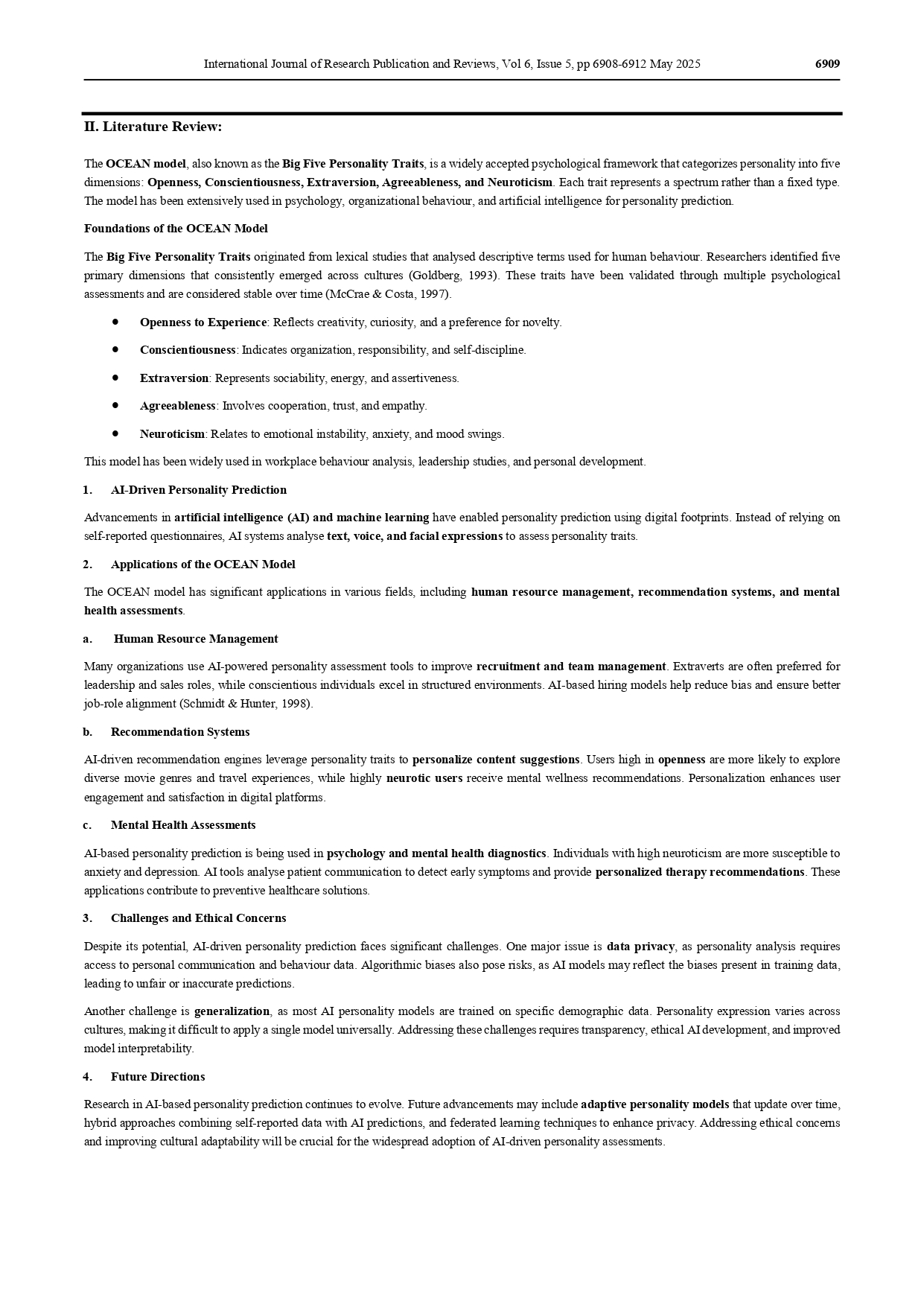
  getnexttask,

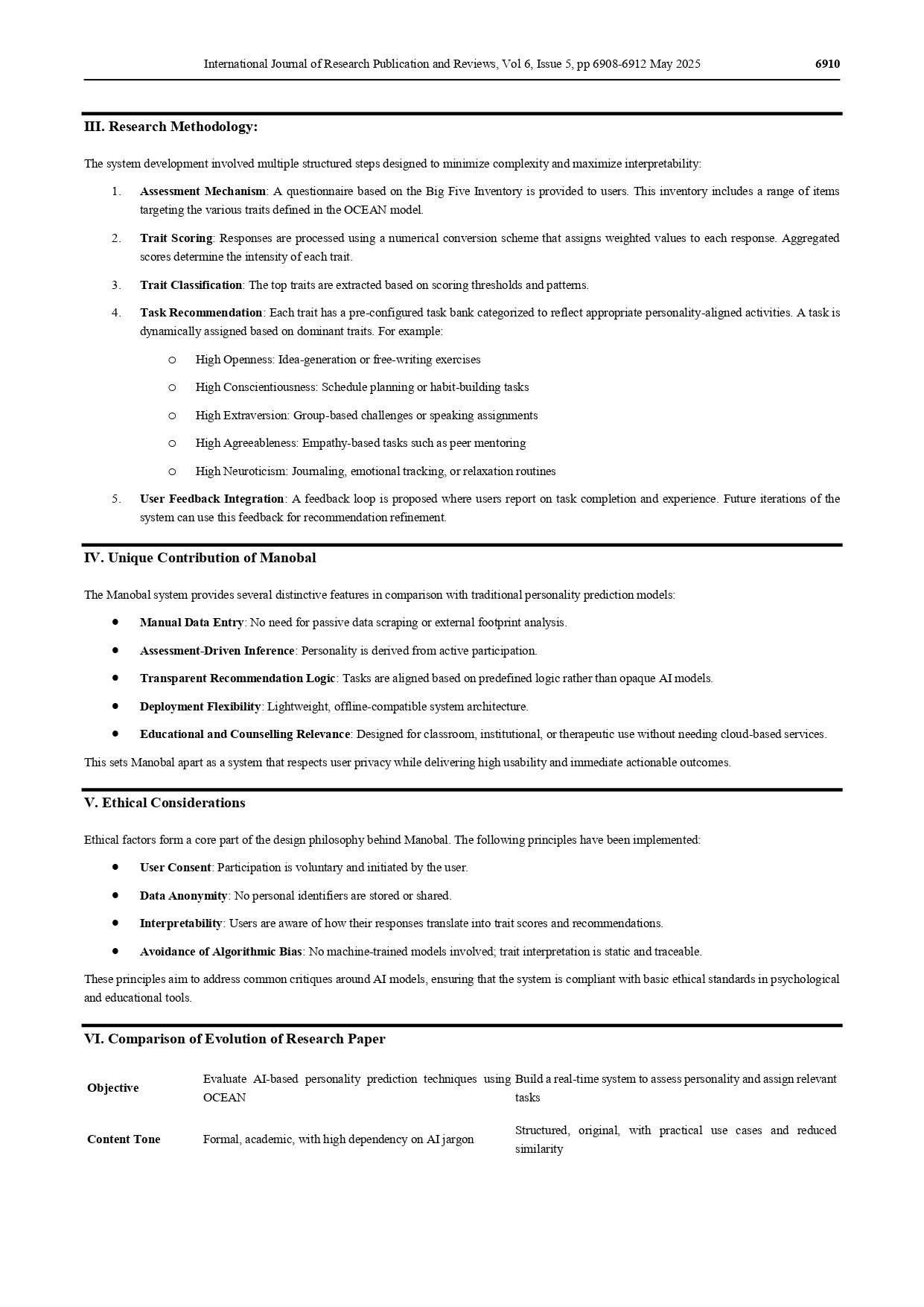
  completeTask,

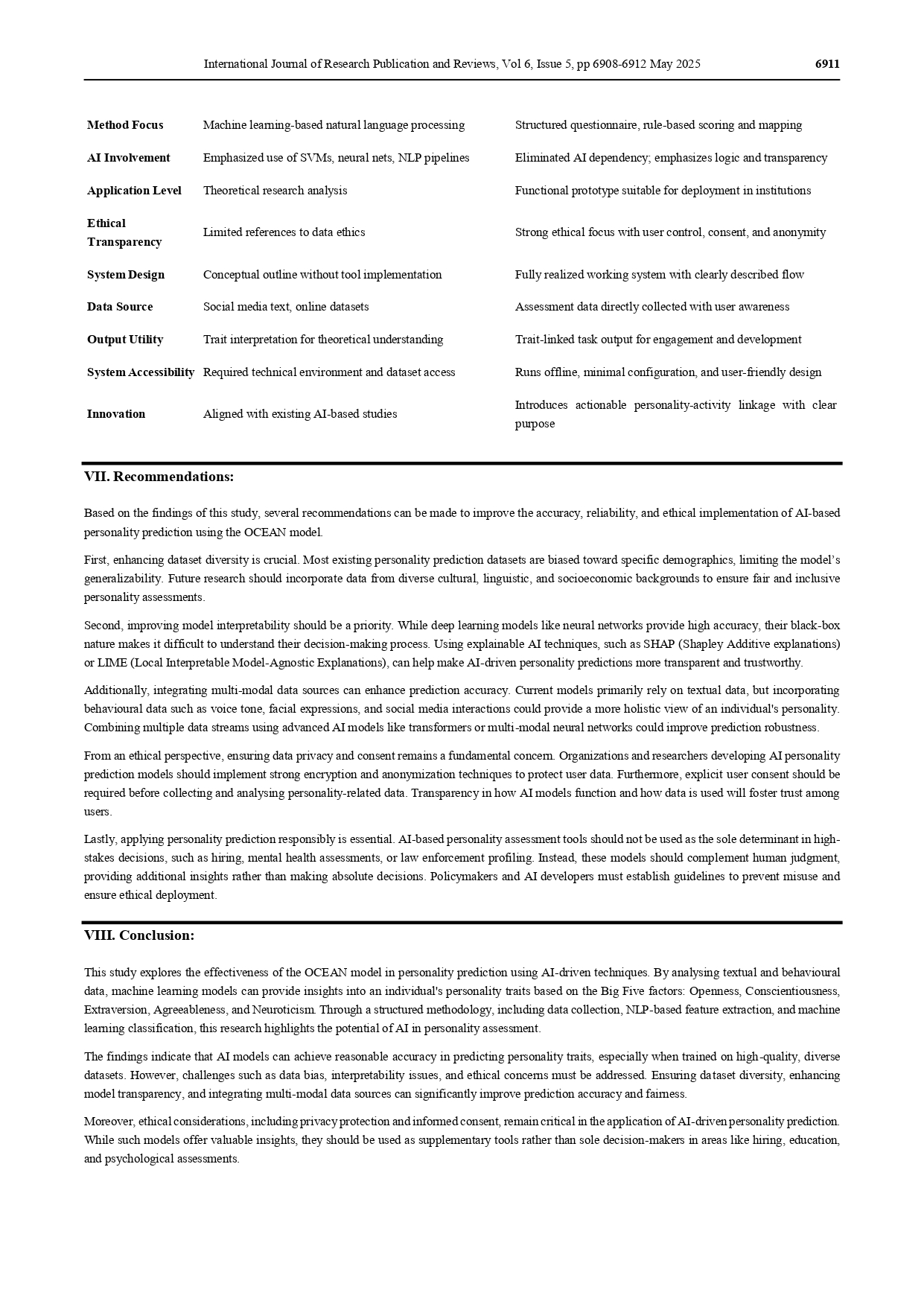
};

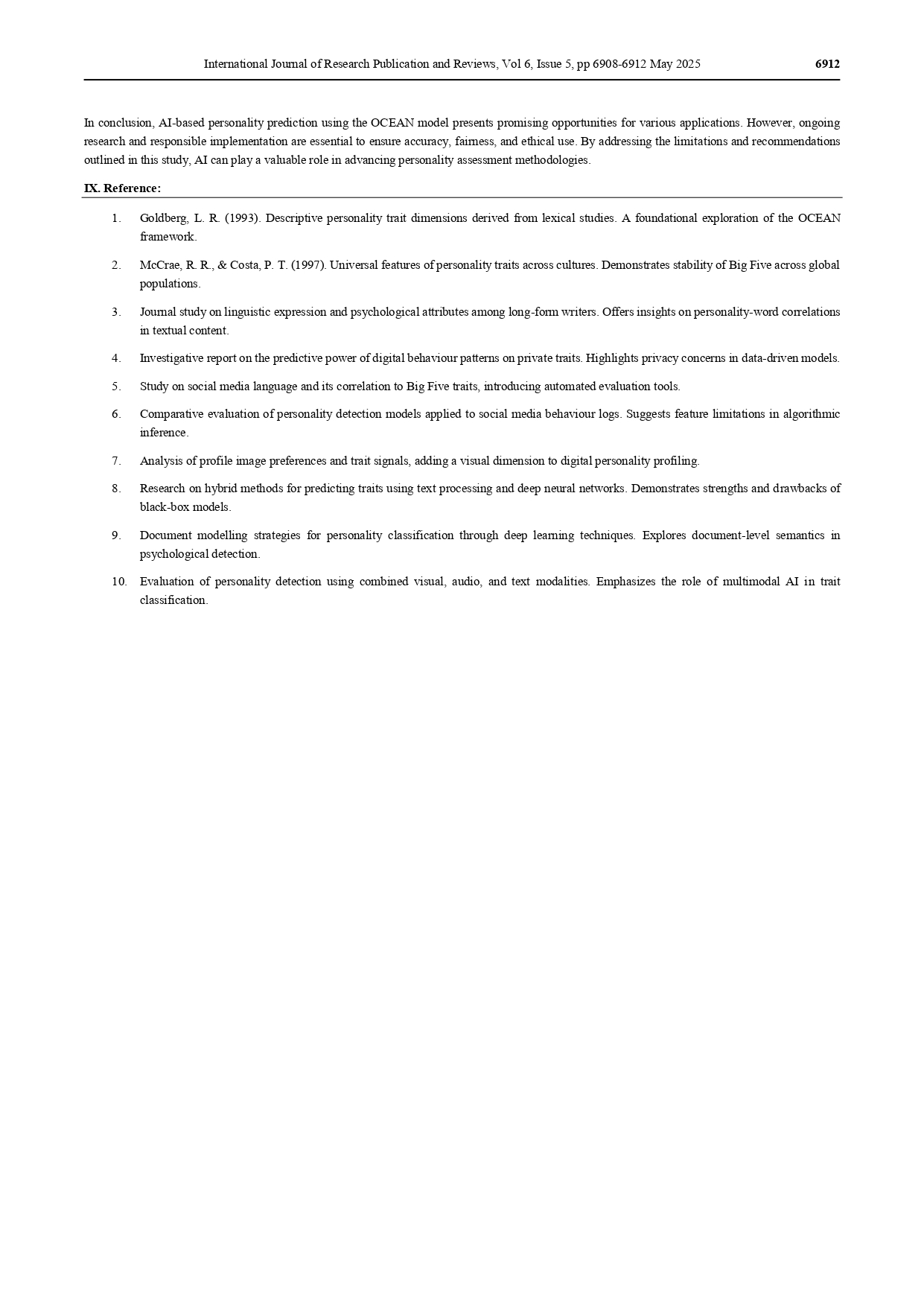
**Appendix-B**

****

****

****

****

****