

**PRIORITY-BASED EXAMINATION TIMETABLE SCHEDULING  
USING ANSWER SET PROGRAMMING APPROACH: A CASE STUDY  
OF COVENANT UNIVERSITY**

**OTUNUYA HENRY CHUKS  
(19PCG02025)**

**SEPTEMBER, 2021**

**PRIORITY-BASED EXAMINATION TIMETABLE SCHEDULING  
USING ANSWER SET PROGRAMMING APPROACH: A CASE STUDY  
OF COVENANT UNIVERSITY**

**BY**

**OTUNUYA HENRY CHUKS**

**(19PCG02025)**

**B.Sc Computer Science, Madonna University, Nigeria**

**A DISSERTATION SUBMITTED TO THE SCHOOL OF  
POSTGRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE  
(M.Sc) DEGREE IN COMPUTER SCIENCE IN THE DEPARTMENT OF  
COMPUTER AND INFORMATION SCIENCES, COLLEGE OF  
SCIENCE AND TECHNOLOGY, COVENANT UNIVERSITY.**

**SEPTEMBER, 2021**

## **ACCEPTANCE**

This is to attest that this dissertation is accepted in partial fulfilment of the requirements for the award of the degree of Master of Science in Computer Science in the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota, Nigeria.

**Mr. John A. Philip**

**(Secretary, School of Postgraduate Studies)**

**Signature and Date**

**Prof. Akan B. Williams**

**(Dean, School of Postgraduate Studies)**

**Signature and Date**

## **DECLARATION**

I, **OTUNUYA HENRY CHUKS (19PCG02025)** declares that this research was carried out by me under the supervision of Prof. Ambrose A. Azeta of the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota, Ogun State, Nigeria. I attest that the dissertation has not been presented either wholly or partially for the award of any degree elsewhere. All sources of data and scholarly information used in this dissertation are duly acknowledged.

**OTUNUYA HENRY CHUKS**

**Signature and Date**

## **CERTIFICATION**

We certify that this dissertation titled “**PRIORITY-BASED EXAMINATION TIMETABLE SCHEDULING USING ANSWER SET PROGRAMMING APPROACH: A CASE STUDY OF COVENANT UNIVERSITY**” is an original research work carried out by **OTUNUYA HENRY CHUKS (19PCG02025)** in the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota, Ogun State, Nigeria under the supervision of Prof. Ambrose A. Azeta. We have examined and found this work acceptable as part of the requirements for the award of Master of Science in Computer Science.

**Prof. Ambrose A. Azeta**

**(Supervisor)**

**Signature and Date**

**Dr. Olufunke O. Oladipupo**

**(Head of Department)**

**Signature and Date**

**Prof. Olusegun Folorunso**

**(External Examiner)**

**Signature and Date**

**Prof. Akan B. Williams**

**(Dean, School of Postgraduate Studies)**

**Signature and Date**

## **DEDICATION**

This research is dedicated to the Almighty God and Jesus Christ my Lord, through and for whom all things were made.

## **ACKNOWLEDGEMENTS**

My utmost gratitude goes to the Almighty and ever-living God, who is the inspiration for this research, and for His divine grace from which I have always benefited. In Him I live, move and have my being.

I sincerely wish to give thanks to my parents, Mr. and Mrs Francis Otunuya Idigbe for the sponsorship, support, and love given to my right from the beginning of my master's programme in Covenant University.

Special thanks goes to my supervisor, Prof. Ambrose A. Azeta for his guidance, corrections and support throughout the duration of this research.

My profound gratitude goes to the head of department, Dr. Olufunke O. Oladipupo, the PG coordinator, Dr. Aderonke A. Oni for her insightful corrections and entire faculty of the Computer and Information Sciences department for their direct and indirect contribution during this programme.

And lastly, I want to thank all my colleagues for their questions and motivation during the entire period of this research work.

## TABLE OF CONTENTS

<b>CONTENT</b>	<b>PAGE</b>
<b>COVER PAGE</b>	
<b>TITLE PAGE</b>	<b>ii</b>
<b>ACCEPTANCE</b>	<b>iii</b>
<b>DECLARATION</b>	<b>iv</b>
<b>CERTIFICATION</b>	<b>v</b>
<b>DEDICATION</b>	<b>vi</b>
<b>ACKNOWLEDGEMENTS</b>	<b>vii</b>
<b>TABLE OF CONTENTS</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF TABLES</b>	<b>xiv</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xvi</b>
<b>ABSTRACT</b>	
<b>CHAPTER ONE: INTRODUCTION</b>	<b>1</b>
1.1 Background of the Study	1
1.2 Statement of the Problem	3
1.3 Aim and Objectives of the Study	3
1.4 Methodology	4
1.5 Significance of the Study	5
1.6 Scope and Limitation of the Study	6
1.7 Definition of Terms	6
1.8 Organization of the Dissertation	7
<b>CHAPTER TWO: LITERATURE REVIEW</b>	<b>8</b>
2.1 Introduction	8
2.2 Answer Set Programming (ASP)	8



2.2.1 Overview	8
2.4 Related Works	17
2.5 Commonalities in Literature	19
2.6 Gaps in Literature	23
<b>CHAPTER THREE: RESEARCH METHODOLOGY</b>	<b>24</b>
3.1 Introduction	24
3.2 An Overview of the Examination Timetabling Problem	24
3.3 Proposed Model for Examination Timetabling	25
3.4 Requirements Specification	26
3.4.1 Functional Requirements	26
3.4.2 Non-functional Requirements	27
3.5 Examination Timetabling Model Formulation	28
3.5.1 Problem Definition	28
3.6 Algorithms Implemented	32
3.7 Requirements Modeling	36
3.7.1 Use Case Modeling	36
3.7.2 Data Modeling	51
3.7.3 Class Diagram	57
3.7.4 Component Diagram	58
3.7.5 Activity Diagram	59
3.7.6 Deployment Diagram	61
3.8 Deployment Architecture	61
<b>CHAPTER FOUR: RESULTS AND DISCUSSION</b>	<b>63</b>
4.1 Introduction	63
4.2 System Requirements	63

4.2.1 Hardware Requirements	63
4.2.2 Software Requirements	63
4.3 Programming Technologies Specification	64
4.4 Implementation Screenshots	64
4.5 A Resulting ASP Program	86
4.6 Performance Analysis of the Examination Timetabling Model	93
<b>CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS</b>	<b>99</b>
5.1 Summary	99
5.2 Conclusion	99
5.3 Contributions to Knowledge	100
5.4 Recommendations	100
<b>REFERENCES</b>	<b>101</b>
<b>APPENDIX A: SOME SOURCE CODES</b>	<b>105</b>
A.1 The Examination Timetable Generation Source Code	105
A.2 The Database Class	116
A.3 Conversion of Exam Dates and Times into ASP Terms	120
A.4 Conversion into ASP Hard Constraints	120
A.5 Source Code for Executing the Model's Instances	121
<b>APPENDIX B: SOME DATA USED BY THE SYSTEM</b>	<b>122</b>
B.1 Exam Date and Time Data	122
B.2 Exam Halls Data	122
<b>APPENDIX C: RESULT OUTPUT</b>	<b>123</b>
C.1 Examination Timetable Visual Outputs	123
<b>APPENDIX D: QUESTIONNAIRE ADMINISTERED</b>	<b>124</b>

## LIST OF FIGURES

FIGURE	TITLE OF FIGURE	PAGE
1.1	The ASP workflow	5
2.1	Workflow for imperative programming	9
2.2	Workflow for ASP	10
2.3	Visual display of the graph colouring stable model	17
2.4	Architecture for solving a curriculum-based course timetabling problem using ASP	22
2.5	Timetabling generation system general view	23
3.1	Proposed architectural model of the examination timetabling problem	25
3.2	Course fact to course database table mapping	31
3.3	Exam room fact to exam hall database table mapping	31
3.4	Fields for registered course record	32
3.5	Use case for student actor	36
3.6	Use case for student 'Manage course' option	37
3.7	Use case for student 'Course result' option	38
3.8	Use case for student 'Examination timetable' option	39
3.9	Use case for admin actor	40
3.10	Use case for admin 'Manage student' option	41
3.11	Required fields for the student list	41
3.12	Use case for admin 'Manage college' option	42
3.13	Use case for admin 'Manage department' option	43
3.14	Use case for admin 'Manage programme' option	44
3.15	Use case for admin 'Manage courses' option	45
3.16	Required fields for the course list	46
3.17	Use case for admin 'Manage results' option	46
3.18	Required fields for the examination results list	47
3.19	Use case for admin 'Manage exam hall' option	48

3.20	Use case for admin 'Manage constraints' option	49
3.21	Use case for admin 'Manage exam timetable' option	49
3.22	Data model of the system	51
3.23	Class diagram for the MYSQLDB class	57
3.24	Component diagram of the system	58
3.25	Activity diagram of the system's login process	59
3.26	Activity diagram of the examination timetable generation process	60
3.27	Deployment diagram of the developed system	61
3.28	Deployment architecture of system	62
4.1	Screenshot of admin login page	65
4.2	Screenshot of admin home page	65
4.3	Screenshot of admin 'Manage students' option	66
4.4	Screenshot of admin 'Manage college' option	67
4.5	Screenshot of admin 'Manage departments' option	67
4.6	Screenshot of admin 'Manage programmes' option	68
4.7	Screenshot of admin 'Manage courses' option	69
4.8	Screenshot of admin 'Manage results' option	69
4.9	Screenshot of admin 'Manage exam halls' option	70
4.10	Screenshot of admin 'Manage constraints' option	71
4.11	Screenshot of admin 'Manage exam timetable' option	72
4.12	Screenshot of admin 'Upload Student List' modal window	72
4.13	Screenshot of admin 'View student' modal window	73
4.14	Screenshot of admin 'Add College' modal window	74
4.15	Screenshot of admin 'Add Department' modal window	74
4.16	Screenshot of admin 'Add Programme' modal window	75
4.17	Screenshot of admin 'Upload Course List' modal window	76
4.18	Screenshot of admin 'Upload Student Result' modal window	77
4.19	Screenshot of admin 'Add Examination Hall' modal window	78

4.20	Screenshot of admin 'Set Constraints' modal window	79
4.21	Screenshot of admin 'Set semester and credit unit limit' modal window	80
4.22	Screenshot of admin 'Set Examination Dates and Times' modal window	80
4.23	Screenshot of admin 'Generate Examination Timetables' modal window	81
4.24	Screenshot of student login page	82
4.25	Screenshot of student home page	82
4.26	Screenshot of student 'Manage course' option	83
4.27	Screenshot of student 'Course result' option	83
4.28	Screenshot of student 'Examination timetable' option	84
4.29	Screenshot of student admin-approved exam timetable	84
4.30	Screenshot of student 'Timetable unavailable' message	85
4.31	Screenshot of student 'Register Course' modal box	85
4.32	Screenshot of student 'View my results' modal box	86
4.33	Bar chart of SUS score of individual respondent	98

## LIST OF TABLES

TABLE	TITLE OF TABLES	PAGE
1.1	Research objectives and methodology mapping	4
2.1	ASP grounders	10
2.2	ASP solvers	11
2.3	Differences between ASP and Prolog	13
2.4	Similarities and differences between ASP and SAT	14
3.1	The encodings of the model and their meanings	29
3.2	Register course use case narrative	37
3.3	View result use case narrative	38
3.4	View examination timetable use case narrative	39
3.5	Upload student list use case narrative	41
3.6	Add new college use case narrative	42
3.7	Add new department use case narrative	43
3.8	Add new programme use case narrative	44
3.9	Upload course list use case narrative	45
3.10	Upload result use case narrative	47
3.11	Add examination hall use case narrative	48
3.12	Set semester and course unit limit use case narrative	50
3.13	Student database table dictionary	52
3.14	College database table dictionary	52
3.15	Course database table dictionary	53
3.16	Course registry database table dictionary	53
3.17	Constraints database table dictionary	54
3.18	Approved timetable database table dictionary	54
3.19	Credit unit limit database table dictionary	54
3.20	Department database table dictionary	55
3.21	Exam hall database table dictionary	55

3.22	Exam result database table dictionary	56
3.23	Exam date and duration database table dictionary	56
3.24	Present semester database table dictionary	56
3.25	Programme database table dictionary	57
4.1	Instance description and execution details	93
4.2	Summary of respondent data	95
4.2	Result of all responses in the 5-point scale	97
4.4	SUS score for each respondent's response	98

## LIST OF ABBREVIATIONS

<b>Abbreviation</b>	<b>Full meaning</b>
<b>ACO</b>	Ant Colony Optimization
<b>ASP</b>	Answer Set Programming
<b>CTD</b>	Course to Day
<b>CTT</b>	Course to Time
<b>CSS</b>	Cascading Stylesheet
<b>CU</b>	Covenant University
<b>CIS</b>	Computer and Information Sciences
<b>CST</b>	College of Science and Technology
<b>CSV</b>	Comma-separated Values
<b>DBMS</b>	Database Management System
<b>ETSP</b>	Examination Timetable Scheduling Problem
<b>EttS</b>	Examination Timetabling System
<b>SAT</b>	Satisfiability solving
<b>GC</b>	Graph Colouring
<b>GA</b>	Genetic Algorithm
<b>HBMO</b>	Honey Bee Mating Optimization
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ILP</b>	Integer Linear Programming
<b>NP</b>	Non-deterministic Polynomial time
<b>PSO</b>	Particle Swarm Optimization
<b>SUS</b>	System Usability Scale
<b>UML</b>	Unified Modeling Language
<b>UI</b>	User Interface
<b>UG</b>	Undergraduate



## ABSTRACT

Timetabling is the process of assigning an event subject to one or more constraints. It is a complex problem by nature. Majority of the scheduling problems known today fall under the NP-hard (NP for non-deterministic polynomial time) class of computational problems. This complexity is as a result of the many events which needs to be efficiently scheduled using limited resources. The examination timetable scheduling problem is one whose main events are examination courses with resources including examination venues, days and time slots. The primary purpose of examination timetable scheduling is to arrange exams in such a way as to avoid clashes bringing about a seamless examination process for students. The complexity of the examination timetable scheduling problem has been addressed using several heuristic approaches like Genetic Algorithm (GA) and Ant Colony Optimization (ACO) algorithm. The objective of this study was to provide a priority-based examination timetable scheduling model using the Answer Set Programming (ASP) approach. ASP unlike conventional techniques which have being employed in solving the timetable scheduling problem, offers a purely declarative approach to problem solving and is designed chiefly for NP-hard computational problems. An ASP examination timetable model was proposed in this research using the departments of Computer and Information Sciences, and Architecture in Covenant University Nigeria used as case study. The model was implemented using the web technologies of clingo-wasm (a clingo WebAssembly), HTML (Hypertext Markup Language), CSS (Cascading Stylesheet), JS (JavaScript) and Node.js. clingo-wasm which is a core component of the system was used for the grounding and solving of the developed ASP encodings. Five (5) instances of the model were analyzed in results session. The results show the execution times taken by each instance with the total time ranging from 1.47 to 7.763 CPU seconds for all the examination timetable solutions (stable models) executed. The proposed model can be adapted for other variants of the examination timetable scheduling problem (ETSP) as its elaboration tolerant feature would make such adaptations easy to do.

**Keywords:** Priority-based scheduling, Answer Set Programming, Examination timetabling, Scheduling, Combinatorial problem