PES
UNIVERSITY

Report on

“Title of the project”

Submitted in partial fulfillment of the requirements for ***Sem VI***

Compiler Design Laboratory

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

<Name 1>	**<SRN 1>**
<Name 2>	**<SRN 2>**
<Name 3>	**<SRN 3>**

Under the guidance of

<Name of the Course Instructor>
<Designation>
PES University, Bengaluru

January – May 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	**INTRODUCTION (Mini-Compiler is built for which language. Provide sample input and output of your project)**	01
2.	**ARCHITECTURE OF LANGUAGE:** ● What all have you handled in terms of syntax and semantics for the chosen language.	02
3.	**LITERATURE SURVEY (if any paper referred or link used)**	03
4.	**CONTEXT FREE GRAMMAR (which you used to implement your project)**	
5.	**DESIGN STRATEGY (used to implement the following)** ● SYMBOL TABLE CREATION ● ABSTRACT SYNTAX TREE ● INTERMEDIATE CODE GENERATION ● CODE OPTIMIZATION ● ERROR HANDLING - strategies and solutions used in your Mini-Compiler implementation (in its scanner, parser, semantic analyzer, and code generator). ● TARGET CODE GENERATION	
6.	**IMPLEMENTATION DETAILS (TOOL AND DATA STRUCTURES USED in order to implement the following):** ● SYMBOL TABLE CREATION ● ABSTRACT SYNTAX TREE (internal representation) ● INTERMEDIATE CODE GENERATION ● CODE OPTIMIZATION ● ERROR HANDLING - strategies and solutions used in your Mini-Compiler implementation (in its scanner, parser, semantic analyzer, and code generator). ● Provide instructions on how to build and run your computer	
7.	**RESULTS AND possible shortcomings of your Mini-Compiler**	
8.	**SNAPSHOTS (of different outputs)**	
9.	**CONCLUSIONS**	
10.	**FURTHER ENHANCEMENTS**	
REFERENCES/BIBLIOGRAPHY		