Main Report Title
Subtitle

Assignment XXX for Course Name. Course AE6969-I of the 2nd year Bachelor Aerospace Engineering 2022/2023, TU Delft. Project group A01 mentored by *M. Entor*.

Executed and written by:

Student One	0000001	**Student Six**	0000006
Student Two	0000002	**Student Seven**	0000007
Student Three	0000003	**Student Eight**	0000008
Student Four	0000004	**Student Nine**	0000009
Student Five	0000005	**Student Ten**	0000010

Submitted for assessment to:
I.N. Structor, M. Entor

TU Delft

Technical University Delft, Faculty of Aerospace Engineering

Kluyverweg 1, 2629 HS Delft
info@tudelft.nl, www.tudelft.nl

Starting date: January 1st 2022 *Completion date: December 30th 2022*

Table of Contents

List of Figures II

List of Tables II

List of Symbols III

Preface 1

Summary 2

1 Introduction 3

2 Literature study 4

3 Design 5

4 Manufacturing 6

5 Testing 7

6 Conclusion 8

A Task distribution 9

Bibliography 9

List of Figures

List of Tables

List of Symbols

Symbol	Definition	Unit
D	Drag	[kN]
L	Lift	[kN]
x, y, z	Cartesian coordinates	[m]
ε	Strain	[-]
σ	Stress	[MPa]

Preface

Summary

PART 1

Introduction

PART 2

Literature study

References [1, 2]

PART 3

Design

PART 4

Manufacturing

PART 5

Testing

PART 6

Conclusion

PART A

Task distribution

```python
import numpy as np

fastener_data = np.array([
  [0, 0, 0.1],
  [1, 0, 0.1],
  [0, 1, 0.1],
  [1, 1, 0.1]
  ])  #x-coord, z-coord, area  [m],[m],[m^2]
    #the diameter of the fastener is inferred from the hole area

t_plate = 0.01 #plate thickness [m]

F = np.array([0.3, 0.56,1]) #force on bearing at (0,0) [N]
print("F =", F, "[N]")

#=====================

#calculate CG
cg = np.array([0,0,0])
tot_area = np.sum(fastener_data[:,2])
for p in fastener_data:
  cg = cg + np.array([p[0], 0, p[1]])*p[2]/tot_area
print("CG =", cg, "[m]")

#big moment
M = np.cross(F,cg) #swapped r*F because cg vector is flipped
print("M =",M, "[Nm]")
print("".join(["="]*25))

#Forces due to moments
Ar_sq = np.sum([((p[0]-cg[0])**2 + (p[1]-cg[2])**2)*p[2] for p in
                                        fastener_data])
F_f = F/len(fastener_data)
F_m = np.zeros((len(fastener_data),3))
n = 0
for p in fastener_data:
  F_m[n] = np.cross(M, [p[0]-cg[0], 0, p[1]-cg[2]]) * p[2]/Ar_sq
  n += 1
print("F_m =\n",F_m, "[N]")

#Total forces
F_tot = np.zeros((len(fastener_data),3))
for i in range(len(fastener_data)):
  F_tot[i] = F + F_m[i]
print("F_tot =\n",F_tot, "[N]")

Exception
__init__
True append
```

Bibliography

[1] Knuth, D. E., "Literate Programming," *The Computer Journal*, Vol. 27, No. 2, 1984, pp. 97–111.

[2] Lesk, M. and Kernighan, B., "Computer Typesetting of Technical Journals on UNIX," *Proceedings of American Federation of Information Processing Societies: 1977 National Computer Conference*, Dallas, Texas, 1977, pp. 879–888.