

Murad Haider {

Computer Science Portfolio

2110 Horseshoe glen circle Folsom, CA 95630

(916)-541-6669 | muradh192001@gmail.com

About

Background:

As a recent graduate with a Bachelor of Science in Computer Science from the University of the Pacific in California. Despite being originally from Palestine, my upbringing in California has given me a unique cross-cultural perspective.

Starting as a Business major, I quickly transitioned to Computer Science after a captivating introduction to coding. Problem-solving and programming became my passion, and I realized that success in computer science comes from dedication and ongoing learning.

Looking ahead, I'm thrilled to share that I'll be pursuing a Masters Degree in Computer Science at the University of the Pacific. This decision aligns perfectly with my goal to deepen my expertise in a field that I'm deeply passionate about. Through this advanced degree, I'm committed to refining my skills, engaging with complex concepts, and contributing to the evolving world of technology.

What am I doing now?

I am currently pursuing a Masters Degree in Computer Science at the University of the Pacific, building upon the foundation of my Bachelor's degree. My academic journey is complemented by my roles in two dynamic positions:

Eco Light Trading Co. - Remote Data Analyst (May 2023 - Present):

Specializing in product analysis, optimization, and sales trends. Leveraging data analytics and Tableau for performance insights and demand alignment.

The Profit Brokers LLC. - Lead Algorithm & Indicator Developer (October 2018 - Present):

Leading diverse indicator development on trading platforms. Crafting automated trading systems for optimized portfolio management using Python, R, C++, and C#.

Computer Science Portfolio

The following will highlight some of the experience I have gained in my current years and it is forever growing. Some of my favorite projects are the mario game and a personal project I worked on to parse through an input file and sort the elements within it.

Major-related Courses

- COMP 051: Intro to Computer Science | C++
- COMP 047: Discrete Math for Computer Science
- COMP 053: Data Structures
- COMP 055: Application Development | Java
- COMP 147: Computing Theory
- COMP 157: Design/Analysis of Algorithms
- COMP 141: Programming Languages | Python
- ECPE 170: Computer Systems & Networks | VM/Linux
- COMP 151: Artificial Intelligence | Python
- COMP 159: Computer Game Technologies | C#
- COMP 173: Operating Systems | C

CS Community Involvement

I am a member of the Association of Engineering students (AES) at the University of the Pacific and we propose solutions to ASUOP in order to find solutions to different projects.



Academic and Programming Skills

C++	VSCode	Quantitative Analysis
C#	Visual Studio	Risk Management
C	PyCharm	Algorithmic Trading
Python	Clion	Indicator Development
Java	Rider	Finance
webGL	Git	Team Leadership
HTML	Github	Project Management
JavaScript	Docker	Ready and Eager to Learn
PHP	GCP	Communication
SQL	MySql	Teamwork
Azure Machine Learning	Tableau	Creativity/Innovation
Pinescript	PowerBI	Problem Solving
ThinkScript	React	
AWS	Angular	
R Studio	TensorFlow	
	PyTorch	
	Keras	
	Kubernetes	
	Microsoft Excel	
	Slack	
	Teams	
	Microsoft Office	
	Unity	
	Unreal Engine	
	Vensim	
	Arena	

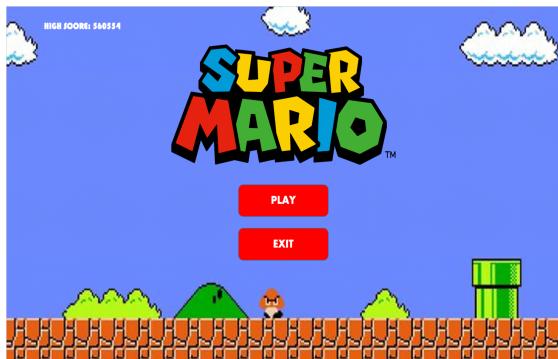
Languages

English

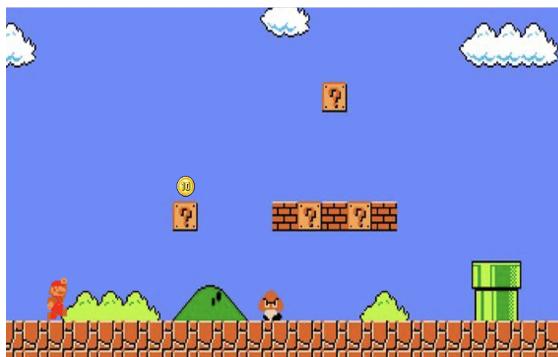
Arabic

Research and Projects

Fall 2020 COMP 055 Application Development Semester Project



For this course, we worked on a game or application of our choice for the duration of the semester. Our only restrictions were that we needed to use Java in eclipse and work with github in order to commit, push, and submit work.



My group and I ended up developing a 2D Tile game remake of Super Mario. This was intended to be a very simple and user-friendly project. We split the roles up and I mainly worked on physics, UI, and collision detection.

Tools Used



Code

This segment of code was particularly the most frustrating and time consuming on my behalf. However, I thankfully had a team so I was not alone in the process. We needed to make sure that the sprite character of our game did not simply go through the entities. We did this by creating hit boxes in order to detect when the entities or objects were being touched. Then we return the hitboxes of the entity passed. We were able to work well as a team in order to get through the collision detection due to the fact we had not done it ever before. This elaborates on my teamwork and problem solving skills.

```
// Detects collision for movement blocking, enemies and
private void detectCollision() {
    for (Entity e : immovable) {
        // Collision between the left side of mainEntity and the right side of immovable objects
        if (getLeftHitbox(mainEntity).intersects(getRightHitbox(e))) {
            mainEntity.setLocation(e.getX() + e.getWidth() + 1, mainEntity.getY());
            mainEntity.xVel = 0;
        }

        // Collision between the right side of mainEntity and the left side of immovable objects
        if (getRightHitbox(mainEntity).intersects(getLeftHitbox(e))) {
            mainEntity.setLocation(e.getX() - mainEntity.getWidth() - 1, mainEntity.getY());
            mainEntity.xVel = 0;
        }

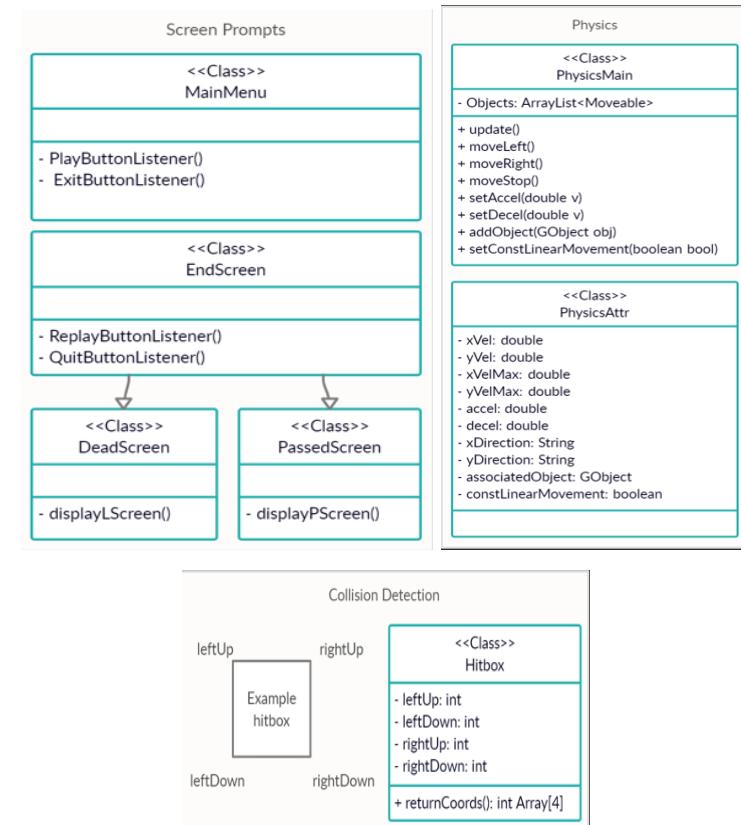
        // Collision between the bottom side of mainEntity and the top side of immovable objects
        if (getBottomHitbox(mainEntity).intersects(getTopHitbox(e))) {
            mainEntity.yDirection = "stop";
            i = 0;
            mainEntity.setLocation(mainEntity.getX(), e.getY() - mainEntity.getHeight() - 1);
        }

        // Collision between the top side of mainEntity and the bottom side of immovable objects
        if (getTopHitbox(mainEntity).intersects(getBottomHitbox(e))) {
            i = 15;
            mainEntity.setLocation(mainEntity.getX(), e.getY() + e.getHeight() + 1);
            mainEntity.yVel = 0;
        }

        if (mainEntity.getX() < 0) {
            mainEntity.setLocation(0, mainEntity.getY());
        }
    }

    // Returns the hitboxes of the Entity passed to it
    public GRectangle getHitbox(Entity ent) {
        return new GRectangle(ent.getX(), ent.getY(), ent.getWidth(), ent.getHeight());
    }
    public GRectangle getTopHitbox(Entity ent) {
        return new GRectangle(ent.getX(), ent.getY(), ent.getWidth(), 5);
    }
    public GRectangle getBottomHitbox(Entity ent) {
        return new GRectangle(ent.getX(), ent.getY() + ent.getHeight() - 5, ent.getWidth(), 5);
    }
    public GRectangle getLeftHitbox(Entity ent) {
        return new GRectangle(ent.getX(), ent.getY(), 5, ent.getHeight());
    }
    public GRectangle getRightHitbox(Entity ent) {
        return new GRectangle(ent.getX() + ent.getWidth() - 5, ent.getY(), 5, ent.getHeight());
    }
}
```

UML Diagram



Presentation Slides



SPRING 2021 COMP 141 Programming Languages Python Parser Project

```
test_input.txt ×
1   z := 0;
2   x := 10;
3   y := 0;
4   while x-y do
5       z := z + y;
6       y := y + 1
7   endwhile
```

For this project I was already very curious about how stock scanners worked and upon doing some research I found there were many ways to parse and sort certain inputs. However, I was not advanced enough to learn how to import all the stock details necessary so this project allowed me to start small with a normal .txt input file and write a program to sort through the file and output the correct identifier for each token.

```
Tokens:
z : IDENTIFIER
:= : SYMBOL
0 : NUMBER
; : SYMBOL
x : IDENTIFIER
:= : SYMBOL
10 : NUMBER
; : SYMBOL
y : IDENTIFIER
:= : SYMBOL
0 : NUMBER
; : SYMBOL
while : KEYWORD
x : IDENTIFIER
- : SYMBOL
y : IDENTIFIER
do : KEYWORD
z : IDENTIFIER
:= : SYMBOL
z : IDENTIFIER
+ : SYMBOL
y : IDENTIFIER
; : SYMBOL
y : IDENTIFIER
:= : SYMBOL
y : IDENTIFIER
+ : SYMBOL
1 : NUMBER
endwhile : KEYWORD
```

The output of the program as you can see went through line by line and sorted each token or element in a line into its correct type.

Tools Used



Code

The following code is the beginning phases of this project. There will be a link to the full project down below. This segment of code shows the different variables that we have and the scanner in order to go through each line and find a match for the correct variable or type. We also have the tokens which will set the items or tokens that matched to the corresponding variable. We used python and repl to create this project and then we would run it using the shell.

```
identifier = re.compile('^(a-z|[A-Z])(a-z|[A-Z]|[0-9])*$')
number = re.compile('^[0-9]+$')
symbol = re.compile('^(\\+|-|\\*|/|\\(|\\)|:=|;)')
keyword = re.compile('(?<![a-zA-Z0-9])(if|then|else|endif|while|do|endwhile|skip)(?![a-zA-Z0-9])')

def scanner(line):
    for item in line:
        exp = ''
        for j in item:
            exp = exp + j
            if (len(exp) != 1 and identifier.match(exp) == None and number.match(exp) == None and symbol.match(exp) == None):
                line.insert(line.index(item), exp[:-1])
                line.insert(line.index(item), item[item.index(exp[-1]):])
                line.remove(item)
                break

    tokens = []
    for item in line:
        if keyword.match(item):
            tokens.append((item, "KEYWORD"))
        elif identifier.match(item):
            tokens.append((item, "IDENTIFIER"))
        elif number.match(item):
            tokens.append((item, "NUMBER"))
        elif symbol.match(item):
            tokens.append((item, "SYMBOL"))
        else:
            tokens.append("ERROR READING TOKENS" + item + "\n")
            break
    return tokens
```

Link

Please [Click Here](#) to view the full project and code.

Senior Project Futures Trading Algorithm

The system architecture consists of a single computer with Windows 10 or higher and NinjaTrader 8. The computer will be connected to a prop firm account, which will allow the user to access the NinjaTrader 8 strategy builder and C# environment. The futures trading algorithm will be designed and run in real-time using historical market data.

The Futures Trading Algorithm will be based on an object-oriented design using UML. The software design will consist of a class diagram showing all classes and their associations, including public and protected members. The main entry/exit condition of the algorithm will be based on the EMA and VMA crosses. These crosses will help determine the direction of the trend and provide signals for entering and exiting trades.

In addition to the EMA and VMA crosses, the algorithm will also incorporate the use of different candle types to reduce the number of trades taken in chop and filter price action. The use of stochastics and the dynamic nature of these indicators will also help to identify potential trades with a higher probability of success.

Furthermore, the algorithm will be designed to run in real-time using historical market data. This allows the trading strategy to be backtested and optimized prior to live trading. By using NinjaTrader 8 and the Rithmic Trader software, the user will have access to a variety of tools for analyzing and testing the performance of the algorithm. The software will be designed to be user-friendly and easy to navigate, making it accessible to traders of all skill levels.

Tools Used



Code

The implementation of the code for this project was carried out using C# within the script editor of the NinjaTrader platform. The link provided below grants access to both the code itself and the associated project's wiki, which outlines the comprehensive steps undertaken to tackle this endeavor. Through the script editor, C# was utilized to construct a robust and dynamic solution, ensuring the project's objectives were met effectively. The wiki offers detailed insights into the iterative approach taken, highlighting the various tests conducted to refine and optimize the codebase. This project stands as a testament to the power of C# within the NinjaTrader environment, showcasing its capability to create intricate and efficient solutions for complex challenges. Access the link for a closer look at both the code and the thought process that drove its development.

Strategy Analyzer			
Display Summary (\$)			
Performance	All trades	Long trades	Short trades
Total net profit	\$325,880.00	\$166,400.00	\$159,480.00
Gross profit	\$335,760.00	\$173,040.00	\$162,720.00
Gross loss	(\$9,880.00)	(\$6,540.00)	(\$3,240.00)
Commission	\$0.00	\$0.00	\$0.00
Profit factor	33.98	26.06	50.22
Max. drawdown	(\$80.00)	(\$80.00)	(\$40.00)
Sharpe ratio	1.25	1.25	1.25
Sorino ratio	1.00	1.00	1.00
Ulcer index	0.00	0.00	0.00
R squared	0.97	0.99	0.99
Probability	0.00%	0.00%	0.00%
Start date	1/1/2022		
End date	5/2/2023		
Total # of trades	5843	3050	2793
Percent profitable	95.77%	94.56%	97.10%
# of winning trades	5595	2884	2712
# of losing trades	247	166	81
# of even trades	0	0	0
Total slippage	0	0	0
Avg. trade	\$55.77	\$54.56	\$57.10
Avg. winning trade	\$60.00	\$60.00	\$60.00
Avg. losing trade	(\$40.00)	(\$40.00)	(\$40.00)
Ratio avg. win / avg. loss	1.50	1.50	1.50
Max. consec. winners	115	123	145
Max. consec. losers	2	2	1
Largest winning trade	\$60.00	\$60.00	\$60.00
Largest losing trade	(\$40.00)	(\$40.00)	(\$40.00)
Avg. # of trades per day	22.27	11.63	10.65
Avg. time in market	0.02 min	0.03 min	0.01 min
Avg. bars in trade	0.04	0.05	0.03
Profit per month	\$26,156.16	\$13,355.79	\$12,800.37
Max. time to recover	2.93 days	2.93 days	0.94 days
Longest flat period	3.99 days	3.99 days	4.00 days
Avg. MAE	\$6.78	\$8.71	\$4.64
Avg. MFE	\$57.56	\$56.88	\$58.32
Avg. ETD	\$1.79	\$2.30	\$1.22

Link

Please [Click Here](#) to view the full project and code.

OHLC Half Cycle of Time and Price

This project introduces a custom technical indicator designed for the financial markets. Developed in collaboration with an experienced Chicago CME market technician, the indicator combines various technical analysis components to provide traders with valuable insights into market dynamics and potential trading opportunities.

Key Features:

Bollinger Bands with Dynamic Standard Deviations: The indicator utilizes Bollinger Bands for technical analysis. The Bollinger Bands employ dynamic standard deviations from the mean. These standard deviations are adjusted as follows:

2.6 Standard Deviations

3 Standard Deviations

3.6 Standard Deviations

4 Standard Deviations

Oscillator: The indicator includes an oscillator that calculates the difference between four different Exponential Moving Averages (EMAs) and the highest high and lowest low over the past 144 bars. This oscillator serves as a robust tool for identifying trends and gauging market momentum.

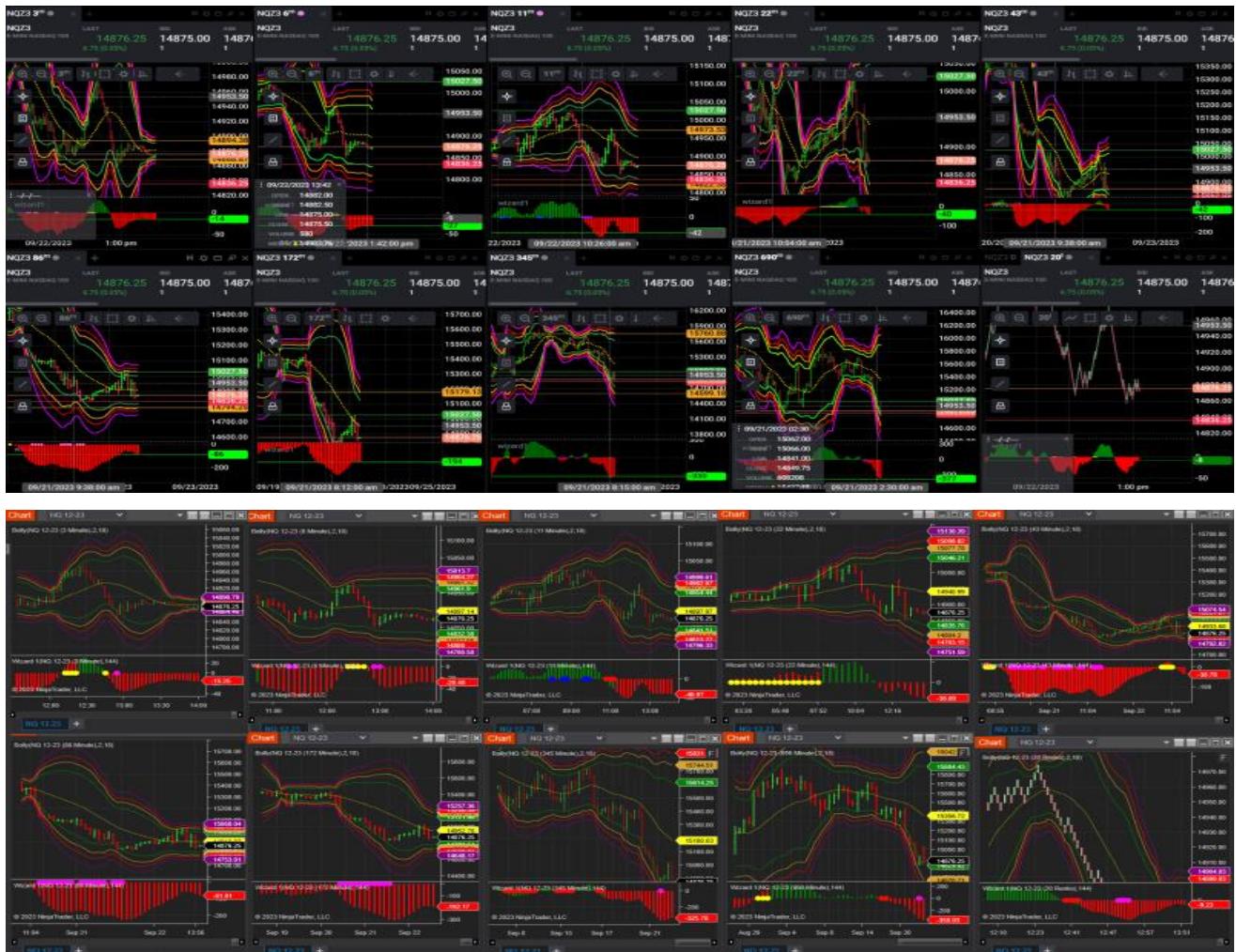
Elliott Wave Pattern Detection: Leveraging chaos theory, the indicator identifies potential Elliott Wave patterns, including "3 up, 3 down" and "4 up, 4 down" sequences. These patterns are invaluable for traders engaged in wave analysis, aiding in precise trend recognition and predictive analytics.

Technical Implementation: The indicator is implemented using JavaScript and C#, ensuring compatibility with a variety of trading platforms and software tools, making it highly accessible for traders across different environments.

Practical Application: Traders and analysts can seamlessly apply this custom indicator to financial charts, enabling them to decode market trends, pinpoint potential trend reversals, and identify wave patterns. It empowers traders to make well-informed decisions and effectively manage risk within their trading strategies.

It should be noted that this should be utilized in conjunction with other analytical techniques and risk management strategies to optimize trading outcomes.

Tools Used



[Link](#)

Please [Click Here](#) to view the full project and code.

Resume

Please [Click Here](#) to view or download a PDF version of my resume.

Murad Haider

(916)-541-6669 | murad192001@gmail.com | <https://www.linkedin.com/muradhaider> | Folsom, California

Education

University of the Pacific, Stockton, CA

- Bachelor of Science, Computer Science
- Concentration in Artificial Intelligence, Graphics and Simulation

Relevant Courses:

- Discrete Math, Data Structures, Programming Languages, Computing Theory, Artificial Intelligence, Computer Game Technologies, Design/Analysis of Algorithms, Web Applications, Computer Simulation, Computer Graphics

Skills

Programming Languages: C++, C#, C, Python, Java, Scala, webGL, HTML, JavaScript, PHP, SQL, Azure Machine Learning, Pinescript, ThinkScript

Software: VSCode, Visual Studio, PyCharm, Clion, Rider, Git, Github, AWS, Docker, GCP, MySQL, Tableau, PowerBI, React, Angular, TensorFlow, PyTorch, Keras, Kubernetes, Microsoft Excel, Slack, Teams, Microsoft Office, Unity, Unreal Engine, Vensim, Arena, R Studio, Agile

Technical Skills: Quantitative Analysis, Accuracy, Algorithmic Trading, Indicator Development, Work Ethic, Team Leadership, Project Management, Ready and Eager to Learn, Communication, Teamwork, Creativity/Innovation, Problem Solving

Languages: English and Arabic

Professional Work Experience

Eco Light Trading Co. - *Remote Data Analyst*

May 2023 - Present

- Specializing in product analysis, optimization, sales record, market trends, and inventory management
- Utilize data analytics techniques and tableau to assess product performance and demand patterns
- Collaborate with cross-functional teams to align product orders with customer demand and optimize stock levels

The Profit Hub LLC. - *Lead Algorithm & Indicator Developer*

October 2019 - Present

- Developed diverse indicators on trading platforms (Pinescript, JavaScript, C#, Python), amplifying market analysis precision
- Engineered automated trading systems (Python, R, C++, C#), optimizing portfolio management and trading strategies
- Led development efforts, adapting algorithms to portfolios, expertise in merging technical acumen with financial strategy

Sarehan Engineering, Ltd. - *Hybrid Software Engineer*

Summer 2023

- Developed software solutions in visual studio tailored to construction and building industry needs
- Collaborated with multidisciplinary teams to streamline project management and enhance efficiency
- Implemented innovative software features to improve project tracking and communication among cross-functional teams

The Cube - *Maker Space IT Technician*

November 2021 - May 2023

- Worked at University maker space to help produce: grant papers, videos, 3D/resin prints, and provide technological support
- Managed hardware and software installations, ensuring optimal functionality of student/staff equipment
- Assisted makers with troubleshooting, fostering a seamless creative environment for innovative projects

Additional Experience

Adams Auto Sales - *Finance Manager*

June 2022 - May 2023

- Communicate with customers & clients to arrange financing and ensure resources & products are according to their needs

Inter Electric - *Electrician*

March 2020 - August 2020

• Utilized blueprint plans to complete wiring systems including circuits, outlets, load centers, and panels

SacCapitalAuto - *Sales Representative*

October 2019 - March 2020

- Exemplified adept salesmanship and product knowledge to consistently exceed targets and deliver tailored automotive solutions

Awards

Senior Project Award In Recognition of Excellence in Engineering Design In Computer Science

- *Futures Trading Algorithm* - This sophisticated system has consistently demonstrated its prowess by achieving an impressive Sharpe ratio of 1.8 and a remarkable Sortino ratio of 2.4. Executing trades with fast speed, resulting in consistent profitability.

Community Involvement & Engagement

Masjid Annur Youth Events, Sacramento, CA

- *Mens Youth Coordinator*; assist in planning events to ensure Muslim youth have an established community and safe space

Professional References

Vincent Klaib

Sac Capital Auto
Financial Manager
3307 Fulton Ave. Sacramento, CA 95821
(916) - 826 - 4544

Christopher Crawford

The Cube - University of the Pacific
Supervisor
3601 Pacific Ave, Stockton, CA 95211
(209) - 479 - 8580

Gustavo Martinez

A1 Auto
Supervisor
3303 Fulton Ave, Sacramento, CA 95821
(916) - 606-2263

Hamode Serhan

Eco Light Trading Co.
Supervisor
0 Alain, Majed Al-Kurum, Israel 2019000
+972 - 054 - 487 - 8486

Ahed Sarehan

Sarehan Engineering LTD.
Supervisor
0 Alain, Majed Al-Kurum, Israel 2019000
+972 - 054 - 480 - 9373