L² LaunchToLead

Resume Analysis & Rewrite

Jacob Rothschild — Computer Engineering

Analyzed against the Impact Bullet Builder™ framework

Overall Score

30

out of 100

0 — Invisible 50 — Achiever 100 — Impact Standard

4/10

Accomplishments

Work exp has some; projects are task lists

5/10

Metrics

Some time savings, 97% cost cut — no $ amounts

6/10

How / Method

Tools shown in context — solid

2/10

Why / Business Impact

"Hours saved" hinted but no CEO-level impact

Verdict: "The Doer" Tier — On the Edge

Jacob's resume is a notch above most new-grad resumes because it has **some real metrics** — 50+ hours saved, 97% cost reduction, 10+ hours saved. That puts it right on the border of "The Achiever" tier. But those numbers are orphaned: they're not connected to a dollar amount, a business outcome, or a problem that was solved. The university projects read like lab reports, and the swim club section is pure filler. With the Impact Bullet Builder formula applied, this resume could jump to 65+ easily — the raw material is strong.

🔍 Key Issues Identified

1 Metrics exist but are disconnected from business impact

"50 plus hours saved annually" and "cutting annual costs by approximately 97%" are good numbers — but 97% of WHAT? $500? $50,000? Without a dollar anchor, the reader can't gauge the impact. The **Why** component of the Impact Bullet Builder is missing on every single bullet. No one reading this knows why the CEO would care.

2 "Made code more maintainable" — vague and uses weak verb

"Made" is one of the weakest verbs you can open a bullet with. And "more maintainable" is a subjective claim with no proof. This violates the **BS Meter** principle — show it, don't claim it. How much more maintainable? What did you refactor? How many lines? How many fewer bugs in the next sprint?

3 "Co-led" and team language throughout projects

"Co-led the project management" and "with a team of students" — this is classic **Me-in-We Extractor™** territory. "Co-led" is hedge language. Did you lead or didn't you? What was YOUR specific contribution to the project management? "Led a collaborative effort to validate" is similarly vague — what did YOU personally validate?

4 No Dragon Slayer context on any bullet

Every bullet jumps straight into the action with no setup. What was broken? What was slow? What was costing money? Per the Dragon Slayer technique — the 97% cost cut is meaningless without knowing what the problem was before Jacob fixed it. Paint the monster, THEN slay it.

5 "Learned how to program" — passive opener

Starting a bullet with "Learned" tells the reader you were a student, not a contributor. The fact that Jacob picked up Verilog in 1 month is impressive — but it needs to be reframed as a result, not a process. "Self-taught Verilog in under 30 days, then programmed..." flips it from passive to assertive.

6 "enhancing functionality and user experience" — filler phrase

"Enhancing functionality and user experience" is a generic phrase that means nothing. This violates the **BS Meter**. What functionality? What user experience specifically? Replace with something concrete: "enabling operators to identify [X] from [X] distance" or "reducing false-positive detection rate by [X]%."

7 Swim club section is pure filler

"Managed budget, collected dues from 50 members" and "Led swim practices once a week" are low-value bullets taking up prime resume real estate. Either rewrite these with measurable leadership impact or cut the section entirely and use the space for stronger technical content.

8 Skills section is a bare list

"Programming Languages: C and Python" and "Tools & Platforms: GitHub and Git" are thin. Per the Impact Bullet Builder: show skills in action, not as a dead list. Jacob mentions Python and multiprocessing in context in the work section — that's good — but the Skills section itself adds almost no value. It should either be enriched with usage context or integrated into bullets.

Rewritten Resume — Impact Bullet Builder™ Applied

[X] = placeholder where Jacob needs to provide missing information (metrics, specifics, or context we can't know from the outside)

Jacob P. Rothschild

(512) 466-9785 | jpr051502@gmail.com | LinkedIn | Lubbock, TX

EDUCATION

Bachelor of Science: Computer Engineering — Aug 2020 – May 2024

Texas Tech University, Lubbock, TX — GPA: 3.88/4.0

Study Abroad: Texas Tech University, San José, Costa Rica (Summer 2022)

WORK EXPERIENCE

Software Engineer Intern — Feb 2025 – Jun 2025

Universal Tech Movement

- REWRITTEN Inherited a job-scraping pipeline that took **[X hours]** to process **[X]** sponsor job boards; re-architected the scripts in Python with multiprocessing, cutting runtime by **[X]%** and saving the team 50+ hours annually — freeing engineering bandwidth for **[X — feature work? other priorities?]**.
- REWRITTEN Refactored **[X — how many?]** Python modules by **[X — what specifically? extracting functions? adding type hints? restructuring classes?]**, reducing codebase complexity by **[X]%** and cutting future update time by 10+ hours — enabling **[X — the next engineer? the team?]** to onboard and ship changes **[X]x** faster.
- REWRITTEN Discovered that the existing pipeline was re-summarizing **[X — all?]** job descriptions on every run via OpenAI's API, costing approximately **$[X]/year**; redesigned the logic to only process new or updated postings, slashing API costs by 97% (from **$[X]** to **$[X]/year**) — directly reducing the company's operating expenses.

- REWRITTEN Built and launched the company's public job board website on WordPress, customizing the HTML, CSS, and PHP template to display **[X]** scraped postings with **[X — search? filters? categories?]** — the site attracted **[X — visitors/month? users? sign-ups?]** within the first **[X]** weeks of launch.

UNIVERSITY PROJECTS

Interactive Pumpkins — Facial Expression Detection System

Aug – Oct 2022

Texas Tech University — Project Lab 2

- REWRITTEN Engineered the embedded software layer for a Raspberry Pi–based facial expression detection system using Python, OpenCV, and DeepFace — the system recognized **[X — how many?]** distinct expressions in real-time and triggered **[X]** unique pumpkin responses with **[X]%** detection accuracy.
- REWRITTEN Designed the software architecture flowchart before writing any code, mapping all edge cases and data paths — this upfront planning eliminated **[X]** integration errors and saved the **[X]**-person team over 5 hours of debugging time.

Robocop — Autonomous Rover with Target Detection

Jan – May 2022

Texas Tech University — Project Lab 1

- REWRITTEN Self-taught Verilog in under 30 days, then programmed a Xilinx Artix-7 FPGA to autonomously navigate a rover chassis along a metallic path using inductive proximity sensors — the rover completed the **[X — course? test track?]** with **[X]%** path-following accuracy.
- REWRITTEN Developed the target detection and strike module in Verilog — the system decoded blinking LED signals, displayed status characters on a seven-segment display, and autonomously fired a projectile when it identified a red LED signal, successfully hitting the target in **[X out of X]** test runs.
- REWRITTEN Owned the integration testing process across **[X]** subsystems, systematically combining and validating components in staged builds — this approach caught **[X]** interface errors early and saved the team 10+ hours of rework time.

LEADERSHIP

Treasurer & Practice Lead

Aug 2021 – May 2024

Texas Tech Swim Club

- CONSIDER REMOVING ~~Managed budget, collected dues from 50 members, and organized fundraisers. Led swim practices once a week. Provided coaching to swimmers by giving feedback on stroke and technique.~~

  → If keeping: "Managed a $[X] annual budget for a 50-member athletic club, collecting dues from all members at a [X]% on-time rate and organizing [X] fundraisers that raised $[X] — funding [X — travel? equipment? event entry fees?]. Led weekly practices and coached [X] swimmers on stroke technique, contributing to [X — any team results? meet placements? membership growth?]."

TECHNICAL SKILLS

Languages: Python (multiprocessing, OpenCV, DeepFace, API integration), C (embedded systems), Verilog (FPGA programming — Xilinx Artix-7)

Hardware: Raspberry Pi, FPGA (Xilinx Artix-7), microcontrollers, electrical schematics, lab instrumentation

Web: WordPress (HTML/CSS/PHP customization), Git & GitHub

Before & After — Best Examples

✗ **ORIGINAL (Doer Level)**

"Optimized Python scripts to summarize job descriptions, through OpenAI's API, only on new or updated postings, cutting annual costs by approximately 97%."

✓ **REWRITTEN (Impact Standard + Dragon Slayer)**

"Discovered that the existing pipeline was re-summarizing all job descriptions on every run via OpenAI's API, costing approximately $[X]/year; redesigned the logic to only process new or updated postings, slashing API costs by 97% (from $[X] to $[X]/year) — directly reducing the company's operating expenses."

✗ **ORIGINAL (Doer Level)**

"Made code more maintainable resulting in 10 plus hours of time savings for future updates."

✓ **REWRITTEN (Impact Standard)**

"Refactored [X] Python modules by [extracting functions / adding type hints / restructuring classes], reducing codebase complexity by [X]% and cutting future update time by 10+ hours — enabling the next engineer to onboard and ship changes [X]x faster."

✗ ORIGINAL (Doer Level)

"Learned how to program a FPGA in Verilog rapidly in 1 month; thereafter, programming a rover chassis to follow a metallic path in Verilog using a Xilinx Artix-7 FPGA and inductive proximity sensors."

✓ REWRITTEN (Impact Standard + Dragon Slayer)

"Self-taught Verilog in under 30 days, then programmed a Xilinx Artix-7 FPGA to autonomously navigate a rover chassis along a metallic path using inductive proximity sensors — the rover completed the [course/test track] with [X]% path-following accuracy."

Summary: What Jacob Needs to Do

1

Anchor the 97% cost cut with a dollar amount

97% is a strong number — but it means nothing without context. Was that savings from $10,000/year to $300/year? Or from $100 to $3? Go back and calculate the actual dollar amount. "Slashed API costs by 97%, saving $[X]/year" is the difference between a good bullet and a great one.

2

Fill in every [X] placeholder above

The rewritten bullets have the structure right — but only Jacob knows the actual numbers. Go through each [X] and plug in the real data. If you don't remember exactly, estimate conservatively. A conservative estimate is infinitely better than no number at all.

3

Add Dragon Slayer context to top 3 bullets

Especially the multiprocessing optimization, the OpenAI cost cut, and the website launch. What was the pain before? How slow was the old pipeline? How much were they spending before you fixed it? What happened when Jacob's website went live vs. how they handled it before? Paint the problem, then the solution.

4

Kill "Co-led," "Made," and "Learned" as openers

Replace every hedge verb. "Co-led" → "Led" (if true) or describe the specific thing you did. "Made" → "Refactored." "Learned" → "Self-taught X, then [did Y]." Own your contributions.

5

Decide: swim club stays or goes

If Jacob can quantify leadership impact (budget size, membership growth, fundraiser revenue, coaching outcomes), keep it and rewrite it. If not, cut it and use that space for an additional technical project, open-source contribution, or the Costa Rica study abroad experience — anything with more substance.

6

Add website launch metrics

Jacob launched a real website that people used. That's impressive — but the bullet doesn't capture any of the results. How many job postings does it show? How many visitors in the first month? Is it still running? Those metrics turn "launched a website" from a task into a business accomplishment.

7

Enrich the skills section with usage context

Instead of "Programming Languages: C and Python" — write "Python (multiprocessing, OpenCV, DeepFace, API integration)" and "Verilog (FPGA programming — Xilinx Artix-7)." Show a recruiter that you've actually USED these tools in real projects, not just listed them.

✓ What's Already Working

Real metrics already present

50+ hours saved, 10+ hours saved, 97% cost reduction, 5 hours saved — Jacob already has the instinct to quantify. The numbers just need dollar anchors and business context to reach Impact Standard.

Strong 3.88 GPA

Near-4.0 GPA in Computer Engineering at Texas Tech is impressive and shows academic rigor. This is a trust signal for early-career roles — keep it prominent.

Real internship with real output

Jacob built and shipped production code — a scraping pipeline, an AI-powered summarization system, and a public website. This isn't "observed meetings" internship filler. It's real engineering work that just needs better framing.

Tools mentioned in context

Python, multiprocessing, OpenAI API, OpenCV, DeepFace, Verilog, Xilinx Artix-7, Raspberry Pi — these all appear naturally within bullet points, not just as a dead list. This is exactly what we teach.

Diverse engineering range

Jacob spans embedded (FPGA/Verilog), computer vision (OpenCV/DeepFace), backend automation (Python/multiprocessing), and web development (WordPress). For an early-career engineer, this breadth is a differentiator — it shows adaptability and fast ramp-up.

Scorecard: Impact Bullet Builder™ Criteria

Criteria	Score	Notes
Accomplishments (not duties)	4/10	Work experience bullets hint at accomplishments (saved time, cut costs, launched something). Project bullets read more like lab write-ups.
Metrics / Quantification	5/10	Several metrics present (50hrs, 10hrs, 97%, 5hrs, 10hrs) — more than most new grads. But no dollar amounts and project sections are metric-free.
How / Method Shown	6/10	Good. Python, multiprocessing, OpenAI API, OpenCV, DeepFace, Verilog, Xilinx FPGA, WordPress all appear in context. Strongest element of the resume.
Why / Business Impact	2/10	"Hours saved" is the closest thing to impact, but no bullet connects to a dollar amount, a business outcome, or a "why the CEO cares" statement.
Me-in-We Extractor™	4/10	"Co-led," "with a team of students," "Led a collaborative effort" — Jacob's specific contributions are blurred by team language in the projects section.
Dragon Slayer Context	1/10	No problem-first framing on any bullet. The 97% cost cut screams for a "before" story but doesn't have one.
Action Verbs	5/10	Mix of strong (Designed, Optimized, Implemented, Launched) and weak (Made, Learned, Co-led, Provided). The weak ones are in prominent positions.
Bullet Quality Consistency	3/10	Wide variance. Work experience is decent, projects are mid-tier, swim club is pure filler. Quality should be uniform across all sections.

Criteria	Score	Notes
OVERALL SCORE	**30/100**	Tier: "The Doer" — bordering Achiever, needs impact framing

The Good News

Jacob has the strongest raw material of most new grads we see: a real software internship where he shipped production code, a 97% cost reduction stat that just needs a dollar anchor, a near-4.0 GPA, and hands-on experience spanning embedded systems, computer vision, and web development. He already quantifies — he just doesn't contextualize. Applying the Impact Bullet Builder formula would likely jump this resume from 30/100 to 70+ in a single session. The foundation is solid — it just needs the framework.