Pocket Mechanics

Nanda Min-Fink, Eva Pavlik, Rey Stone, Sabir Saklayen, Sami Ul-Ahmed

What is Pocket Mechanics?

Pocket Mechanics' Auto Part Finder is a feature rich, easy-to-use website for all your auto part needs. By creating an account on our website, you have the ability to save your vehicles in your garage and search for parts that are guaranteed to be compatible with your vehicle. Our website allows you to save addresses of local stores as well as your home address, making it seamless to get parts shipped and delivered to you.

Features included: Register/Login; Discover page with filtering; Garage with user's vehicles; Account page; Cart; Checkout; Nearby repair shop locator.

Tools

- GitHub Project Board
  - Repository, collaboration, PR's, merging branches
  - Ranking: 4

- PostgreSQL
  - Architecture diagram, organizing initial website ideas and vision
  - Ranking: 4

- VS Code
  - Remote repositories, working on individual parts of project
  - Ranking: 5

- Mocha (testing)
  - Running tests
  - Ranking: 3

- Handlebars
  - Templating, allows reuse of templates for overall website page
  - Ranking: 4

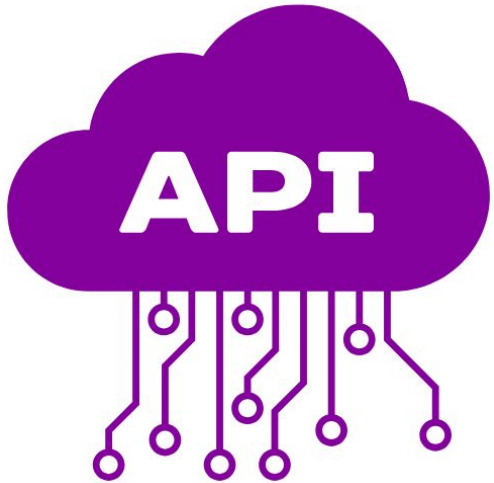
- NodeJS/Express
  - Simplifies web and application building; routing, handling requests
  - Ranking: 5

- Render
  - Online cloud deployment, high-quality images
  - Ranking: 5

- Figma
  - Wireframing
  - Ranking: 5

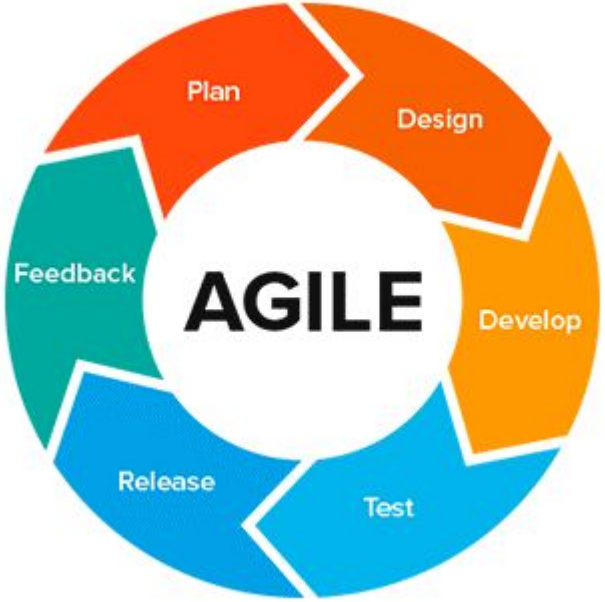
APIs

- Stripe
  - Secure payment system
  - Credit/debit card validation
  - Ranking: 4
- OpenStreetMap
  - Nomatism for address completion
  - Coordinate to address conversion
  - Overpass for finding local shops
  - Ranking: 4
- Rock Auto
  - Technically not api but scraping
  - Used to populate parts/vehicle/pricing tables
  - Ranking: 5



Agile Methodology

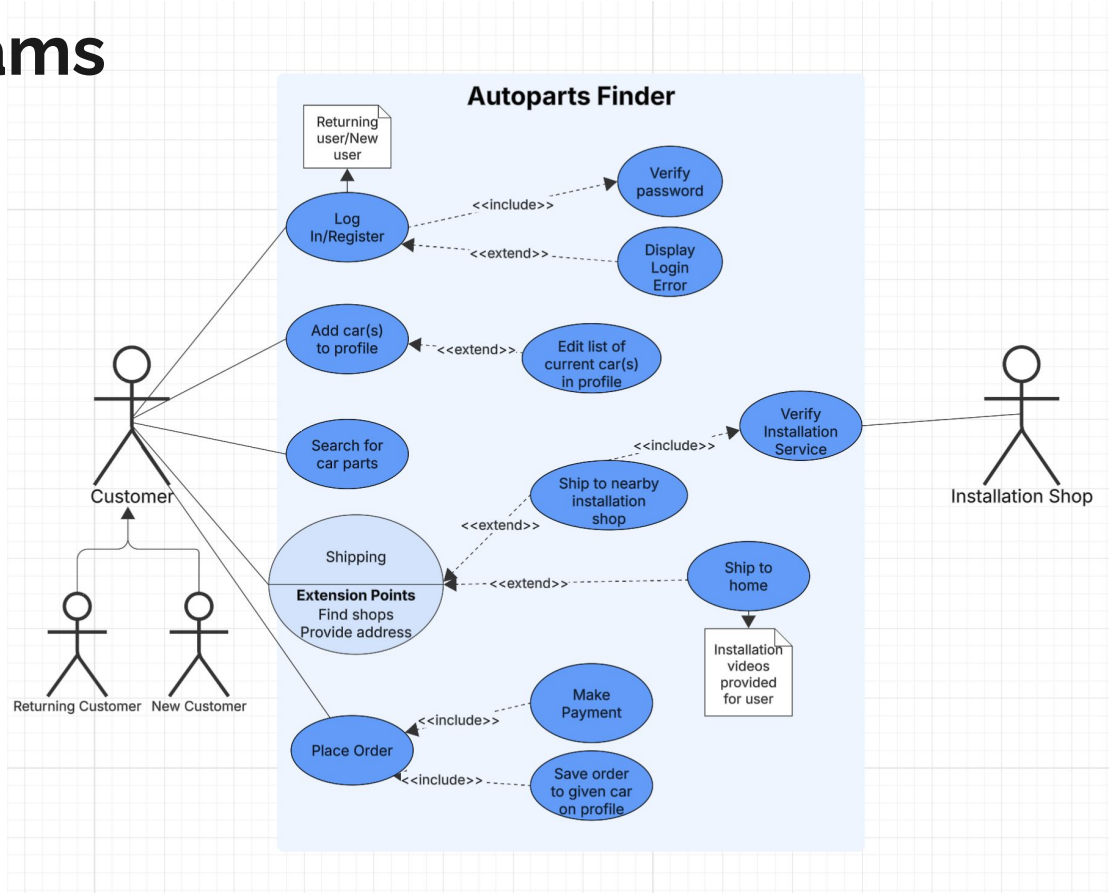
- Defined specific project goals
- Breakdown of tasks and milestones
- Assigned tasks and responsibilities
- Created timeline and schedule
- Monitored progress and adjusted goals



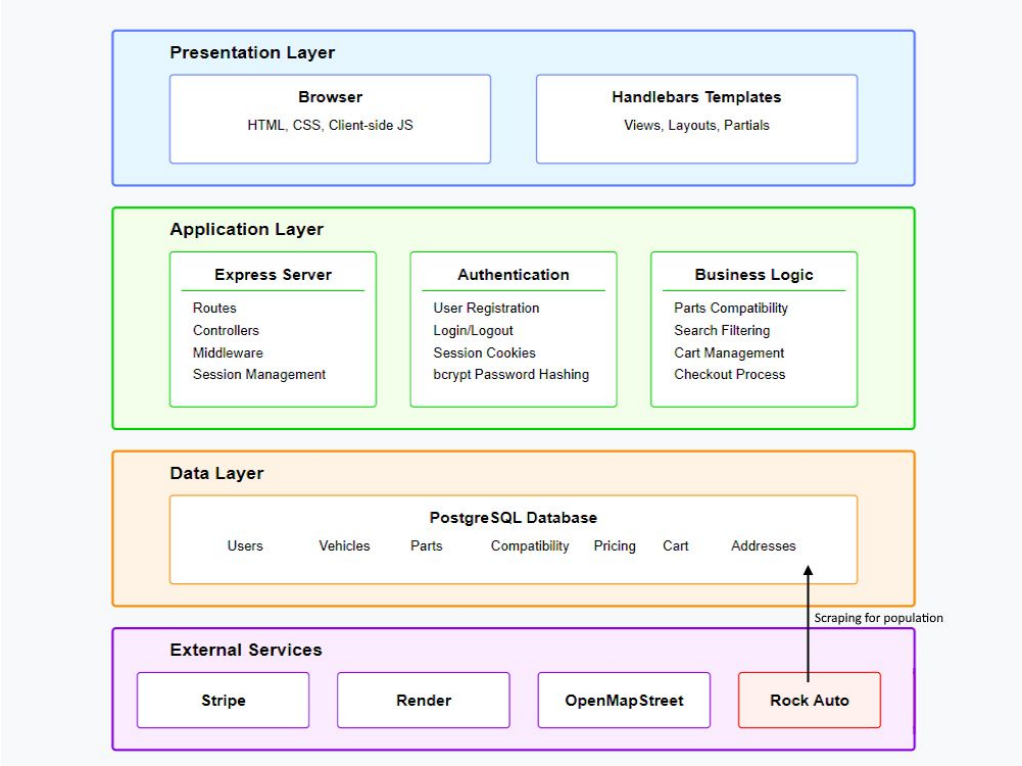
Architecture Diagrams

Features:

- Log in
- Add cars
  - Edit/Delete cars
- Search for parts
- Add to cart
- Shipping
  - To home or nearby shop
- Payment



Architecture Diagrams (cont.)



Challenges

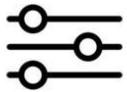
- Gathering data
  - Scraping data due to lack of free resources
    - Time consuming
    - Difficult to automate
  - Introduced rotating proxy server to mask IP address
- Cleaning data
  - Many solutions were space-consuming
    - Limited by storage and RAM on Random PaaS
  - Took sample set of data collected
  - Vehicle compatibility as JSON object embedded as field for each part
- Merge conflicts
  - More merge conflicts as codebase grew
  - Checking old commits, manually inserting new code rather than letting Git manage merge conflicts

Challenges cont.

- Limited resources
  - Affected our design, specifically size of our database
  - Limited time
    - Encouraged use of AGI; helpful but played a role in poor organization
  - Solution: Craft a high-level prototype given our timeframe
- Testing
  - Adapting tests to an early stage changing codebase
  - Comprehensively testing each feature

Enhancements

- Including all car makes/models
- Including more parts
- Implementing at-home installation videos
- Estimation of installation costs
- More filter flexibility
- Implementing Amazon, Ebay APIs
- Order history
- Orders: returns, refunds, exchanges, cancellations
- Current discounts



Live Demo!!

Questions?