**Project Proposal- Intelligent Browser**

**Introduction**

Name: Juyee Shirkhedkar

Roll Number: B24CI1018

I am a second year student at IIT Jodhpur, and my branch is Civil and Infrastructure Engineering. I have been coding since 11th under the board curriculum.

I have the knowledge of basics of Python. Currently, I'm working with natural language processing tools in Python, including tokenization, stemming, and vectorization, as part of my AI/ML learning.

Contact:

- Email- [b24ci1018@iitj.ac.in](mailto:b24ci1018@iitj.ac.in)

- GitHub- username- **js10112006**

- Contact Details- 9511272594

- LinkedIn- <https://www.linkedin.com/in/juyee-shirkhedkar-861817327/>

**Why You’re Interested**

I find this project interesting, and will help me make a way through AI/ML, and more of the backend processing. This will help me learn about the integration of frontend and backend into a fledge dashboard. The functions it puts forward are exciting.

**Strengths**

I know the basics of Python, JavaScript and a bit of integration of NLP. I have tried making API calls from Gemini and integrating it with Python to help find intents. I’m interested in learning more Backend, but I have also worked on HTML, CSS and a bit of PHP.

**What You Want to Learn**

I’m curious to learn about the libraries Puppeteer and Playwright used for Automation of browser, integrating PostgreSQL for the data and working on more JS driven backend.

**Time Commitment**

I can commit up to 5 hours, or more as per the project needs, since right now I am not occupied with any other projects or workload. After the session starts, I might have some other commitments, but I will try to dedicate as much time as possible.

**Approach**

I plan to begin by contributing to the backend, and ensure the efficient functionality of the same. Starting with Natural Language processing and Automation unit will be more efficient and will help build foundation strong.

NLP Commands-

Strating the implementation of Tokenization, Stemming or Vectorisation in order to convert the user commands in to intents to help further processing

Automation- using puppeteer/ playwright to connect the parse intents to the browser and carrying out the display of the particular web automatically.

Later, after building a backend, moving towards creating Frontend Dashboard would be better as per me.

**Project Understanding**

The Project IntelliBrowse aims at making a user friendly, AI- driven browser, which helps in efficient and automated execution of tasks. This helps in reducing the complexity and simplifies the user experience with a hands-free browsing technology, over a user input command. The browser takes care of encryption and password credentials, making it secure. It is capable of reading captchas, navigating through different websites and simplifying the whole experience.

Workflow

* Planning and Designing
* Setting up Libraries and Functions and understanding/ learning them
* Building the Core
* Natural Language Processing by integration API calls from Gemini to help the system understand the input.
* Automated Browser- using puppeteer/playwright to launch different browser as instructed and carry out functions like log in, filling forms and submitting.
* Captcha solving commands if necessary.
* Dashboard which consists of the past data and shows options to start a new task, also displays the status of the task processing.
* Encryption of the credentials to carry out and store secure log in and payments process.
* Testing the Browser
* Deployment

User to output processing

User

↓

[ Text / Voice Command ]

↓

Frontend (Next.js and Tailwind)

> Input box / Mic

> Status Dashboard (Live Task Updates)

> Login Pages

↓

API Call to Backend (Next.js API Routes)

↓

AI/NLP Engine

> Gemini API parses

↓

Backend Logic (Node.js / TypeScript)

> Task Planner

> Uses Puppeteer / Playwright for:

- Opening website

- Logging in

- Filling forms

- Clicking buttons

- Payment (via secure input)

↓

Secure Vault (AES-256)

> Stores:

- Encrypted credentials

- Payment details

- Task history

- API keys

↓

Dashboard Update and display of task status

↓

Deployment (AWS + Docker + CI/CD)

> Runs containers with app & DB

> Automatically deploys from Git repo