



*Young professionals in a new city engaging in a fun local event.*

# AI-Powered Activity Recommender: Competitive Review & Project Plan

## Introduction

Moving to a new city can leave young professionals wondering *"I'm bored, what should I do?"* This hackathon project aims to build an AI-driven **"boredom buster"** web app that acts as a personal agent to recommend activities and events. The agent will learn about users through a brief Q&A (e.g. interests, goals, budget) and by leveraging public online data (such as users' public social media likes or posts). Based on this profile, it will search the web for local happenings – from concerts and meetups to outdoor adventures – and present a curated selection of things to do. The focus is on **major U.S. metro areas** (e.g. DC/Northern VA, NYC, LA), where there's a vibrant variety of events but newcomers might struggle to find what suits them. By providing personalized suggestions *and* easy ways to act on them (links to tickets, directions, etc.), the agent serves as a virtual concierge for newcomers. The following sections provide a **competitive review of similar solutions** in the market, and a detailed **technical plan and business plan** for bringing this idea to life in a weekend hackathon setting.

## Competitive Review of Similar Products

Several products and services already tackle the "what to do when bored" problem, though none *exactly* replicate our agentic, personalized approach. Below is a review of both **AI-powered assistants** and **traditional event discovery platforms** that address this user need:

- **Bored AI (YesChat)** – An AI chatbot designed to alleviate boredom with tailored entertainment suggestions. Bored AI focuses on categories like video games, movies, and outdoor activities, asking about the user's interests and location to recommend options. For example, if a user is interested in outdoor fun, Bored AI will inquire about their location and preferred activity level, then suggest nearby adventures like hikes or city scavenger hunts <sup>1</sup>. This shows the power of personalization (it even considers weather/season) and confirms interest in AI-driven suggestions for leisure. However, Bored AI's scope is broad (including at-home entertainment) and it may not directly integrate real-time local event listings – a gap our project aims to fill by actively searching current events.
- **YesChat "Local Event Finder"** – Another AI tool on the YesChat platform, described as an *"AI-driven guide"* to discover local events matching a user's interests and location <sup>2</sup>. It promises to **curate a personalized list of events** for the user, eliminating the need to manually scroll through event listings <sup>2</sup>. This concept is very similar to our project's core idea. It indicates that AI can be used to streamline event discovery, but details on its implementation are sparse. Our approach could differentiate by building a richer user profile (via Q&A and social media) rather than relying solely on user-selected filters.

- **Eventure AI (concept)** – An upcoming mobile app (in development) focused on suggesting local events based on a user's *mood* and interests <sup>3</sup>. The creator noted that existing platforms like Google, Eventbrite, and Facebook felt “*generic and time-consuming*,” and envisioned an AI that might say, “*Feeling adventurous? Here are hiking groups, night markets, or local concerts happening tonight.*” <sup>4</sup>. This underscores a perceived market gap: people want **contextual, mood-based recommendations** instead of static lists. It's a validation of our project's value proposition. Eventure AI is still early-stage, so there's room for us to stand out by actually building and demoing a working prototype at the hackathon.
- **Eventbrite** – A well-known event marketplace that doubles as a discovery app for local happenings. Users can search by location and date to find anything from cooking classes to concerts. Eventbrite's strength is the sheer volume of events and its filtering options (by category, date, price, etc.), and it even provides location-based recommendations and a “free events” filter <sup>5</sup>. However, **Eventbrite requires the user to do the searching** – it's not personalized unless you manually set filters. Our agent could leverage Eventbrite's data via API, but add value by automatically querying it based on the user's interests (saving the user the work). In competitive terms, Eventbrite is a key source of event information; partnering with or pulling data from Eventbrite can bootstrap our content, while our AI layer differentiates the experience.
- **Meetup** – A platform oriented around interest-based groups and events. It's great for finding niche communities (tech talks, hiking clubs, book clubs, etc.) and is popular among young professionals looking to meet people. Meetup organizes events by groups and interests; for example, popular categories include outdoor adventures, professional networking, or creative workshops <sup>6</sup>. While Meetup is excellent for community-building, it doesn't proactively suggest events to an individual – you have to join groups or search by category. In our competitive landscape, Meetup addresses the *social* aspect of finding things to do (connecting with like-minded people), which is an angle our agent can acknowledge (perhaps by suggesting Meetup events if they match the user's profile). A limitation of Meetup is its fragmented nature (many groups on different sites); our agent could unify recommendations across Meetup and other platforms.
- **Laparizone** – A newcomer (launched 2024) in event discovery focusing on **hyper-local** events. It lets users filter events by very specific interests (from pottery workshops to indie music gigs) in their area <sup>7</sup>. Laparizone's ease of use and fine-grained interest filters show a demand for personalization. Essentially, it's an event listing site with a modern interface and notification features. While it's not AI-driven per se, it's a direct competitor as an alternative to Facebook Events, aiming to ensure you “never miss an event” by enabling category notifications <sup>8</sup>. Laparizone's approach validates that letting users define their interests leads to more relevant suggestions. Our agent can achieve a similar outcome but with *less manual input* from the user by inferring interests through conversation and public data.
- **Nextdoor & Local Subreddits** – These community platforms are indirect competitors in that they help users discover local happenings through crowdsourced content. **Nextdoor** is a neighborhood-focused network where people post local events (yard sales, community gatherings) with a very personal touch <sup>9</sup> <sup>10</sup>. **Reddit** has city-specific subreddits (e.g. r/nyc, r/LosAngeles) where weekly threads often highlight upcoming events and activities <sup>11</sup>. These sources can surface “hidden gem” events that bigger platforms might miss (like a local block party or a niche hobby meetup). For our agent, tapping into these sources (perhaps by scraping a weekly “What's happening” Reddit thread

or Nextdoor public posts) could enrich recommendations. On the flip side, because these rely on user-generated posts, they can be hit-or-miss and not personalized to an individual's interests. Our solution's competitive edge is using AI to filter that noise and present only the relevant gems to each user.

- **Traditional Search & Guides** – Many people still turn to **Google searches** or city guide websites (TimeOut, local news event calendars) when bored. For example, Googling “things to do in [city] this weekend” will yield lists from various sources. City tourism boards maintain calendars of festivals, concerts, etc., and news apps like *Patch* compile community event listings <sup>12</sup>. These are valuable sources of data (and our agent could pull from them), but they are not tailored – everyone sees the same list. Similarly, digital assistants like Siri, Alexa or Google Assistant can answer “What are some events near me?” with generic results, but they don’t have an extended conversation to learn your personal taste. This status quo emphasizes **why** a personalized agent is compelling. In short, there’s plenty of event information out there, but **no single service that intelligently personalizes and aggregates it across sources** in a conversational experience. That is the niche our hackathon project will fill.

**Key Takeaway:** Competitors range from AI chatbots for boredom (focusing on general entertainment) to event discovery websites and community forums. Each addresses part of the problem (finding things to do) but either requires manual effort or lacks personalization. Our solution differentiates itself by combining **breadth of data** (searching across multiple event sources) with **personalization via AI** (learning the user’s preferences automatically). We aim to offer the convenience of an AI assistant with the richness of all the event info on the web.

## Technical Plan (Hackathon Implementation)

**Project Overview:** We will develop a **web-only application** (accessible via browser) using a Python Flask backend. The core will be an AI **Agent** powered by a large language model (LLM) that utilizes tools (via LangChain) to gather information and answer the user. The system flow is: a user interacts with a chat interface on the website → the AI agent asks a few onboarding questions to learn about the user → the agent performs web searches and data fetching to find relevant activities → the agent presents a curated list of suggestions, each with actionable details (e.g. links to event pages or directions).

To deliver this in a hackathon (4 people, ~8 hours each = ~32 person-hours), we’ve scoped the project to be **feasible yet impactful**, focusing on a functional prototype with core features:

### Tech Stack & Architecture

- **Backend:** Python Flask will serve the web app, chosen for its simplicity and our team’s familiarity. Flask will handle routing, serve the HTML/JS for the interface, and provide API endpoints that the front-end can call to get AI responses. We’ll likely use Flask’s built-in dev server during the hackathon, as scaling and deployment are not a concern for the demo. The backend will integrate with the AI and handle external API calls (to event sources, etc.).
- **AI Engine:** The agent will be built using **LangChain**, which allows chaining LLM calls with tool usage. We plan to use the OpenAI API (GPT-3.5 Turbo or GPT-4, depending on availability) for natural

language understanding and generation. LangChain is ideal here because it provides out-of-the-box tools for things like web search and knowledge retrieval, and it makes it easy to switch out components (for instance, swapping one LLM for another) <sup>13</sup>. By using LangChain's agent framework, the AI can decide when to ask the user a question, when to search the web, and when to formulate an answer. For example, LangChain's Web Browser Tool can give the agent the ability to visit websites and extract information from them <sup>13</sup> – we will leverage this to have the agent automatically scrape event listings or social media pages when needed.

- **Frontend:** A simple, mobile-friendly web interface will be created (HTML/CSS/JavaScript). Given time constraints, we might start with a basic Bootstrap template or a lightweight front-end framework (or even just jQuery for simplicity) to create a chat-like interaction. The UI will include a chat dialogue area (to display the agent's questions and suggestions) and input box for the user. We'll also have a results section for the **suggested activities** with clear formatting (each suggestion showing title, short description, maybe an image or icon, and a link/button to proceed). One team member will focus on making the interface clean and easy to use, since **readability and user experience are key**.
- **Database:** For a hackathon prototype, we likely don't need a persistent database. User sessions can be kept in memory (or simple JSON files) just to store conversation state and any retrieved profile info. If needed, Flask's session or a lightweight in-memory store (like Python dict or Redis if available) can maintain state. Long-term, a database might store user profiles or past suggestions, but it's out of scope for the hackathon MVP.

## Key Features & Components

1. **User Profile Elicitation:** When a user first interacts, the agent will ask a few quick questions to understand them. We'll focus on:
  2. **Interests/Hobbies:** e.g. *"Do you enjoy outdoor activities, art, music, or something else?"*
  3. **Goal or Mood:** e.g. *"Are you looking to meet new people, learn something, or just relax and have fun?"*
  4. **Budget:** e.g. *"Any budget or preference for free vs. paid activities?"*
  5. **Location:** If not provided via the interface, the agent will ask the user's city or zip code (limited to our target metro areas for now). Location is crucial since all suggestions will be region-specific.
6. We will also include an **optional social media lookup**: The user can choose to provide a public social profile handle (say their Twitter username). If given, the agent (via backend code) will fetch recent public posts or bio from that profile to infer interests (for example, if their Twitter feed often mentions "hiking" or "Marvel movies"). *Only public data will be used*, respecting privacy and avoiding any login requirements. This feature might be implemented using an API (e.g. Twitter's API for recent tweets) or a quick web scrape if an API isn't readily available at hackathon (taking care not to violate terms). If time is short, we will prioritize the Q&A approach and treat social lookup as a stretch goal.
7. **Web Search for Events:** The agent will use tools to search the internet for suitable activities once it has the user profile. This might involve multiple sub-tasks:
  8. Querying known **event aggregators** (e.g. Eventbrite's public events search API, or Meetup's API if available) for events in the user's city that match their interests (keywords like "tech meetup", "live

- music”, “hiking”, etc.). For instance, if the user indicated interest in music and a low budget, the agent might search Eventbrite for free concerts in NYC this weekend.
9. Searching general websites: We can utilize a search engine API (like SerpAPI or an open Google Search tool) via LangChain to find things like “<City> + <interest> + this weekend”. This could surface results from local newspapers, Reddit threads, or city-specific blogs listing “10 things to do in <City>”.
  10. The agent will likely need to perform **scraping/parsing** on the fly. With LangChain’s browser tool or a custom scraping function, it can click into a result and extract key info (event name, date/time, location, brief description). For the hackathon, we might narrow this to a couple of sources for reliability – e.g., focus on Eventbrite and one community site per city – to avoid too much unpredictable parsing.
  11. We will implement safeguards so the agent doesn’t stray too far or get caught in irrelevant web content. Since time is limited, we might pre-define some specific search prompts for each city/interest to guide the agent. Another approach is using an API aggregator like **PredictHQ** (which provides a unified events feed <sup>14</sup>) if we can get a free trial – but integration might be heavy for a hackathon, so likely we stick with simpler queries.
  12. **Suggestion Generation:** After gathering a pool of candidate activities/events, the LLM will **curate and summarize recommendations** for the user. This involves:
    13. Filtering out anything that doesn’t match the user’s criteria (e.g., if user said “no expensive stuff” and an event has a \$100 ticket, skip it).
    14. Possibly categorizing suggestions (e.g., one social meetup, one outdoorsy thing, one cultural event) to give a diverse mix – this makes the experience engaging (“Here are a few ideas: 1) A networking happy hour tomorrow, 2) A hiking trip on Saturday morning, 3) A free museum night event...”).
    15. Generating a friendly description for each suggestion, explaining why it fits the user: *“This Jazz Night concert matches your interest in live music and is free to attend, just a 10-minute walk from you <sup>5</sup>!”* – including such personal touches can greatly increase relevance. We will use the LLM’s strength in natural language to make the suggestions appealing and customized.
  16. Ensuring each suggestion has an **actionable next step**: a link to RSVP or buy tickets (if available via Eventbrite or official site), a map link for location, or instructions like “just show up at the venue”. This addresses the “easy ways to start activities” goal – the user shouldn’t have to do much else to actually *do* the activity.
  17. **Interactive Refinement:** One advantage of an agentic setup is that the user can have a dialogue. We plan to allow follow-up interaction such as:
    18. User can ask for alternatives or adjustments: *“Hmm, what about something more active?”* or *“Any suggestions for Sunday instead?”* The agent will then refine its search and suggestions accordingly.
    19. The agent might proactively ask for clarification if needed: *“Would you like to do this alone or in a group? I can suggest group activities if you want company.”* – making the experience feel like a conversation with a helpful local guide.
  20. Given the hackathon time, we will implement basic follow-ups (perhaps one round of refinement) to demonstrate this capability. We will harness LangChain’s memory or session context so the agent remembers user preferences stated earlier in the conversation.

21. **City Focus Filter:** Since we are focusing on specific metro areas (DC, NYC, LA, etc.), we will integrate a quick way for the user or system to set the city context. For instance, during onboarding the user may pick from a dropdown of supported cities or the agent will deduce from their input (if they say “I live in Manhattan” the agent knows it’s NYC). This constraint actually helps technically – we can pre-gather or cache some event sources for those cities to speed up the searches. Each city has known go-to sources (e.g. **TimeOut New York** for NYC events, **LA Weekly** for LA, etc.). If time permits before the hackathon, team members might compile a cheat-sheet of URLs or RSS feeds to scrape for each city, to avoid solely relying on live search during the demo (which can be slow or unpredictable).

## Development Breakdown & Feasibility

With 4 team members over ~8 hours each, we will split responsibilities and work in parallel as much as possible:

- **Person 1 (AI/Backend Engineer):** Focus on setting up the LangChain agent and integrating the OpenAI API. This includes defining the prompt templates for the agent (for example, system prompt that guides it to act as a friendly local expert, and a plan for tool usage), and adding tools: one for web search/scraping and possibly a custom tool for fetching social media data. They will test the agent logic in an isolated environment (perhaps Jupyter or unit tests) to ensure it can take a query like “Find me something to do” and go through steps to return an answer. Using LangChain should accelerate this – it has built-in patterns for initializing an agent with tools <sup>15</sup> <sup>13</sup> .
- **Person 2 (Data/API Integrations):** Work on connecting to external data sources. This includes writing functions to call Eventbrite’s API or perform an HTTP request to a known events page and parse results. If Eventbrite API is used, this person will handle authentication keys and formulate API queries (e.g., location radius, categories). They will also implement the optional Twitter/Instagram scraping function. Essentially, Person 2 provides the *tools* Person 1’s agent will invoke. For example, writing a `search_events(city, keywords)` function that returns a few event snippets. Early on, this person and Person 1 will coordinate so that the agent’s tool usage aligns with what Person 2 is implementing (e.g., defining a LangChain tool that calls `search_events` ).
- **Person 3 (Frontend Developer):** Build the Flask routes and HTML/CSS/JS. They will create the main page template with a chat interface and results display. Key tasks include designing a simple **chat bubble** layout for the conversation, an input form that triggers the backend (maybe via AJAX/fetch so the conversation feels dynamic without full page reloads), and displaying the final suggestions in a user-friendly format (could be cards or a list with nice typography). They’ll also handle any city-selection UI and ensure the app is responsive (works on mobile, since young professionals may try this on their phones).
- **Person 4 (Full-stack/Coordinator):** This person will float between tasks, assist where needed, and ensure integration goes smoothly. Initially, they may help Person 3 with UI or Person 2 with writing parsing logic. They will also perform testing as components come together – for example, simulating user interactions to see if the agent responds correctly. Importantly, this role will keep an eye on the time and priority: making sure that by mid-hackathon we have a minimal end-to-end demo (even if some parts are stubbed or simplified), and then adding enhancements. If all core features are done

early, Person 4 might work on bonus features (like a more advanced recommendation algorithm or adding an image to each event from an API).

Given the tight timeline, we will use an **agile approach**: start with the simplest working pipeline and iterate. For instance, a basic milestone by hour ~4: *User inputs "I'm bored" → agent returns one hardcoded suggestion (just to prove the UI and API call work)*. Next milestone: agent uses one source (Eventbrite) live to get a real suggestion. Then add more sources, then add profile Q&A, and so on. This way, we always have something to demo even if we run out of time on later features.

We anticipate challenges in integrating multiple systems quickly (LLM + external APIs + custom logic). To mitigate risk, we plan fallback options: e.g., if the live search fails or takes too long, the agent can fall back to a pre-prepared list of top events per city so the user isn't left empty-handed. Caching some data for the demo (especially if internet at the hackathon is unreliable) is a wise safety net.

In terms of **LangChain model switching**, our architecture makes it easy to swap the LLM or tools. For example, if OpenAI API limits kick in, we could switch to an open-source LLM (like local GPT4All) by just changing the LangChain `ChatModel` configuration, since we abstracted calls via the library. Similarly, if we refine our data source approach after the hackathon, we can plug in a different events API without overhauling the agent logic – it just calls a "find events" tool that we can rewire under the hood. This flexibility is by design, considering future work beyond the hackathon.

## Business Plan

Even though this is a hackathon project, we envision potential for a real product that serves young newcomers in cities. Below we outline the business aspects: **target market, value proposition, competitive advantage, monetization, and go-to-market strategy**.

### Target Audience & Market Opportunity

The primary audience is **young professionals (20s and 30s) who have recently moved to a new city**. This cohort often has disposable income and a desire to explore and socialize, but they lack local knowledge and networks initially. In the US alone, millions of people relocate each year – in 2022, state-to-state movers increased to about 8.2 million <sup>16</sup>, and a large portion of these movers are Gen Z and Millennials <sup>17</sup> (the generations comprising young professionals). This indicates a substantial and growing user base who could benefit from a personalized "what to do" assistant. Initially, we will focus on major urban areas like NYC, DC, and LA, which attract many young professionals and have rich event scenes. These cities provide ample content (lots of events) and high concentrations of tech-savvy users likely to try an AI solution.

User persona example: *Alex, 26, moves to NYC for a new job. It's Saturday, and Alex knows the city has tons to offer but doesn't know where to start. Traditional event sites are overwhelming, and Alex doesn't have a friend group yet to suggest activities.* Our AI agent would be the go-to companion for Alex to find a fun museum event or a meetup with fellow young professionals, tailored to Alex's interests. Success with users like Alex in one city can lead to word-of-mouth and expansion to other cities.

## Value Proposition

For users, the value proposition is **“Your personal AI city guide – never be bored in a new place”**. In more detail:

- **Personalized Recommendations:** Unlike generic event listings, our agent delivers suggestions that align with *your* interests and mood. It’s the difference between scrolling through dozens of unrelated events versus immediately seeing 3-5 you’ll love. This saves users time and makes it more likely they’ll actually go out and do something they enjoy.
- **Comprehensive Search:** The agent aggregates options from across the web – it’s like using Eventbrite, Meetup, Reddit, and local newspapers all at once, without the user having to visit each. This breadth means users won’t miss out on great opportunities simply because they didn’t know where to look. For instance, a user might not think to check the local library’s site for free workshops, but our agent could surface that if it matches their interests (libraries often host community events <sup>18</sup>).
- **Conversational Convenience:** Chatting with the agent is as easy as texting a friend. This lowers the barrier for users who might not be sure what they want – they can express something vague like “I feel like doing something outdoors with maybe some music” and let the agent figure it out. It’s a more natural interaction pattern for a generation used to messaging.
- **Immediate Action:** By providing direct links or next steps (booking, RSVP, directions), we solve the follow-through problem. Users are more likely to actually attend an event if the path from discovery to action is smooth. Our app would let them go from *“I’m bored”* to *“I have plans!”* in minutes.

For event organizers or businesses, our platform could also offer value by driving attendance. This is a secondary aspect but worth noting – if we gain users, organizers will want their events recommended. This could lead to partnerships or listings down the line.

## Competitive Advantage

Our solution’s competitive edge lies in the **integration of AI-driven personalization with real-time local data**:

- **Learning the User:** Competing event apps require manual input (you filter by category or join groups). We proactively learn about the user (even peeking at public social cues) to create a richer profile. This dynamic profiling means even if two users are in the same city, they get different suggestions tuned to them. That level of personalization is not present in incumbents like Eventbrite or Meetup, which treat you more or less as a generic user until you manually personalize your feed.
- **Agentic Automation:** The use of an *agent* (autonomous AI) to handle the heavy lifting (searching multiple sites, reading through event descriptions, comparing with user preferences) is novel. Other solutions might have recommendation algorithms, but those are usually siloed (e.g., Meetup might recommend groups based on what you joined, Eventbrite might show popular events). Our agent goes out and fetches info on-demand across sources. It behaves like a smart assistant who can answer *follow-up questions*, a capability static recommendation systems lack. This interactivity is a huge differentiator – it feels like a conversation with a knowledgeable local friend rather than using a “database”.
- **Breadth + Depth of Content:** As noted in the competitive review, alternatives typically focus on one pool of events (only events on their platform, or only one type of event). By aggregating, we combine breadth (all sorts of events: music, sports, classes, social gatherings) with depth (details about each event). For example, Nextdoor might have hyper-local events but miss big concerts; Eventbrite has big events but misses hyper-local meetups. We aim to capture both. Eventful tried aggregating sources in the past <sup>19</sup>, but without personalization it could still overwhelm the user with irrelevant items. Our AI curation on top of aggregation is a key innovation.
- **Focus on New-in-Town Experience:** We’re not just listing events; we’re solving an emotional need – combating boredom and loneliness for newcomers. Our messaging and features (like asking if the user wants to meet people) are tailored to that scenario, which competitors haven’t explicitly done. This focus can help guide product



decisions (e.g., maybe we add a feature to “invite a coworker” to an event, to help users socialize). It’s a niche but sizable segment that we understand well, giving us a chance to build loyalty among those users.

## Monetization Strategy

In the hackathon, monetization isn’t a primary concern, but for a business plan we have several avenues to consider:

- **Affiliate Commissions:** Many event ticketing platforms (Eventbrite, Ticketmaster, etc.) have affiliate programs or referral fees. If our app directs a user to purchase a ticket, we could earn a small cut. Over many users and events, this can add up. For example, recommending a popular concert via Ticketmaster could yield a percentage of the ticket sale.
- **Sponsored Listings:** Once we have an audience, event organizers or venues could pay to have their event recommended or highlighted (clearly marked as “sponsored”). The AI could incorporate sponsored suggestions if they still match the user’s criteria (to maintain trust). For instance, a local cooking school might pay us to promote their weekend class to users interested in food/drink activities.
- **Premium Subscription:** We could offer a free tier with basic functionality and a **premium tier** with extras. Premium features might include: notifications of upcoming events tailored to you (so you never have to even ask), ability to “follow” certain types of events (e.g., always notify me of jazz concerts in my area), or concierge services like the agent will actually book/reserve spots on your behalf. Given young professionals often pay for convenience (e.g., meal kits, personal trainers), they might pay a few dollars a month for a “social life assistant” that constantly keeps them busy with minimal effort on their part.
- **Data Insights & Partnerships:** Aggregated, anonymized data about what activities young professionals in different cities are interested in could be valuable to marketers, city planners, or tourism boards. We must be careful with privacy, but trends (e.g., “50% of new NYC users show interest in outdoor activities”) could inform businesses. Partnerships could be formed with relocation services or employers; for example, companies hiring interns/new grads could bundle our app as a perk to help those employees settle into the city (improving retention and satisfaction).
- **Advertising:** Least preferable (as it could ruin user experience), but contextual ads (like Uber or Lyft offering a coupon if you go to an event venue, or a sports store advertising on our app if the user’s interested in hiking events) could be a supplementary revenue stream. We’d want to keep any ads very relevant and probably integrate them as subtle suggestions rather than banners.

Our initial plan would likely combine **affiliate revenue** and **pilot partnerships**. For example, in DC/Northern VA, we could partner with apartment complexes that have lots of young tenants or newcomer networking groups, offering the app as a value-add, and in return perhaps those partners promote us.

## Go-to-Market Strategy

To launch this product beyond the hackathon, we would start with a **city-by-city approach**:

- **Pilot in One City:** Start in Washington, DC (for example) where our team has familiarity. Populate the agent with high-quality data sources for that city and maybe manually verify a batch of upcoming events to ensure good recommendations initially. Get a small group of target users (perhaps through local young professional Facebook or Slack groups) to beta test. This helps refine the agent’s performance and the UI based on real feedback.
- **Marketing to Newcomers:** Position the app on channels that newcomers use. This could include advertising on relocation blogs, posting on r/MovingtoNYC or similar subreddits, partnering with HR departments of companies (e.g., give out info during new employee orientation in big firms), or even physical posters at co-working spaces and popular young professional apartment buildings (“Bored in the city? Scan this QR code to find something fun to do within minutes!”).
- **Leverage Social Media:** Encourage users to share their experiences (perhaps a built-in “Share your plan” button after they pick an event). If

someone uses the app to find a cool event, their Instagram story about it can indirectly promote the app. We can also create city-specific social media profiles that regularly post “Top 3 things to do this weekend” – essentially content marketing that establishes us as experts in the local event scene and drives people to the app for personalized picks. - **Network Effects:** If successful, our user base might naturally drive growth – friends inviting friends to events via the app, etc. We could facilitate this by adding features like “**Invite a friend**” (which sends the event info to a friend along with a note “Let’s go – I found this on BoredomBusterAI!”). This not only spreads awareness but also taps into the fact that people are more likely to try something new if a friend suggests it.

Initially, we’d operate in a few major cities where the concept is most needed. As we gather data and revenue, we’d expand to other cities or even to **travelers** (the same idea could apply to someone visiting a city for a weekend, not just those who moved there). Scaling to new locations mainly involves integrating new data sources and possibly tweaking the model if needed for local nuances.

## Future Extensions

In the course of the hackathon, we’ll get a prototype up, but it’s important to note potential future features that can make this a sustainable product: - **Mobile App:** A dedicated mobile app (iOS/Android) can leverage push notifications (“It’s 5 PM on Friday – how about these ideas for tonight?”) and location services (maybe detecting if the user is in a new part of town and suggesting something nearby). Many young professionals practically live on their phones, so a native app could increase engagement and convenience. - **Enhanced AI Capabilities:** We could incorporate user feedback loops – for example, the agent asks afterwards “Did you enjoy the event?” to learn the user’s tastes better. Over time, the AI could become more of a “*lifestyle coach*,” not only reacting to boredom but proactively suggesting things to enrich the user’s social life. Integration with calendar (auto-blocking time for an event you accept) or with friend lists (suggesting to invite someone if you often attend events together) could deepen the experience. - **Community & Social Features:** While one of our differentiators is the single-player experience (no need to have a social network to get suggestions), we could still build a community around the app. Users might opt-in to a feature to see *who else* is interested in an event or find a buddy to go with. This starts edging into Meetup’s territory but with an AI matchmaker twist (“We found 2 other new-in-town folks interested in the Jazz festival; would you like to connect with them?”). This could greatly enhance the value for newcomers looking to make friends. Of course, it introduces moderation and privacy considerations, so it’d be a later-stage idea. - **Monetization Expansion:** Down the line, if we have a strong user base, we might explore organizing our *own* events (using data to identify gaps in what people want) or partnering with city tourism boards to create special experiences for new residents – adding another revenue stream and reinforcing our brand as the go-to for local engagement.

## Potential Risks and Mitigations

- **Cold Start / Content Gaps:** In less active cities or off-season times, there might not be many events to recommend, making the app less useful. We mitigate this by also suggesting general activities (like “visit this popular park” or “try this trending restaurant”) so that even if formal events are few, the user still gets ideas. The AI can generate creative suggestions (“How about a personal walking tour of the art murals in your neighborhood?”) to fill gaps.
- **Accuracy & Trust:** Relying on AI and live data means sometimes the agent might suggest an event that’s outdated or a bit off-mark. This could erode user trust if not handled. We will implement basic verification (checking event dates, avoiding events far outside the user’s stated area) and clearly

present info so the user can verify (e.g., always showing the event date/time and source). As a backup, providing multiple suggestions increases chance at least one is appealing/accurate. During the hackathon demo, we'll likely curate some results to avoid this risk, but for real product we'd constantly refine the scraping and use user feedback to correct errors.

- **Privacy Concerns:** Scraping a user's public profile might raise eyebrows. We address this by making it opt-in and transparent. We only use *publicly available* info and we could even show the user "Here's what I learned about you from Twitter: you tweet a lot about food and tech." so they feel in control. No personal or private data is accessed. Over time, building trust that the AI is a helpful companion (and not a data harvester) is crucial. We'll have a privacy policy explaining exactly what data sources we use and how.
- **Competition Response:** If our idea gains traction, big players could emulate it (e.g., Eventbrite adding a chat assistant, or Google improving its event recommendations). Our defense will be our head start on personalization and the user base loyalty. Also, by being an independent aggregator, we can remain platform-agnostic (recommending events from anywhere) which a single-platform competitor cannot easily do. We'd aim to grow quickly in user satisfaction so that even if others launch similar AI features, we're recognized as the specialist in the "I'm bored" use case.

## Conclusion

In summary, this hackathon project is more than just a fun experiment – it has the ingredients of a viable product for a real market need. We reviewed how current solutions either require users to sift through generic lists or lack a personal touch, and we plan to fill that gap with an AI-driven, Flask-based web app that **actively learns about the user and scours the web to find the perfect activities** for them. Technically, by leveraging frameworks like LangChain and focusing our scope (just web, a few cities, public data), we can achieve a working demo within the hackathon's constraints.

The business plan shows promise: young newcomers are actively looking for ways to engage with their new city, and they have money and motivation to use a service that simplifies that search. By starting with this niche (young professionals in major cities) and delivering real value – measured in **weekends not spent bored at home** – we can grow a user base and explore monetization through partnerships and premium features.

Ultimately, success will be if our agent can turn a user's "I'm bored" into "This sounds exciting, let's do it!". With the competitive landscape mapped and a solid technical plan, our team is set to build an **"Agentic experience"** that could genuinely enrich users' social lives, one personalized suggestion at a time. We're excited to bring this idea to life at the hackathon and hopefully lay the groundwork for a product that extends well beyond it.

### Sources:

- Joscor, A. (2025). *10 Alternatives to Facebook for Local Event Discovery* – Discusses event discovery platforms like Laparizone, Eventbrite, Meetup, Nextdoor, etc., emphasizing interest filters and hyper-local focus <sup>7</sup> <sup>5</sup> .
- YesChat AI. *Bored AI: Your Personalized Entertainment Guide* – Describes an AI assistant that asks for user preferences (games, movies, outdoor activities) and suggests tailored options, even planning outdoor adventures based on user location and interests <sup>1</sup> <sup>20</sup> .

- Lopez Diaz, J. *Local Event Finder: AI Guide to Local Activities* – An AI tool that curates a personalized list of nearby events so you “spend less time searching and more time enjoying,” highlighting the value of automated event curation <sup>2</sup> .
- Reddit (2024). *Eventure AI – discover local events by mood* – User post about an AI-driven events app idea, noting traditional platforms feel generic and pitching mood-based suggestions like hikes or concerts for when you’re “feeling adventurous” <sup>4</sup> .
- U.S. Census Bureau via Raleigh Realty (2025). *Moving Statistics {2025}* – Reports that state-to-state movers rose to 8.2 million in 2022, with most movers being Gen Z and Millennials seeking new opportunities <sup>16</sup> <sup>17</sup> .

---

<sup>1</sup> <sup>20</sup> Bored AI-Free, Personalized Entertainment Finder

<https://www.yeschat.ai/gpts-9t55QeItER8-Bored-AI>

<sup>2</sup> Local Event Finder: Your Ultimate Guide to Local Activities

<https://jarvis.cx/tools/gpts/local-event-finder-50715>

<sup>3</sup> <sup>4</sup> Eventure AI — an app for discovering local events based on your mood : r/SideProject

[https://www.reddit.com/r/SideProject/comments/1jzoifx/eventure\\_ai\\_an\\_app\\_for\\_discovering\\_local\\_events/](https://www.reddit.com/r/SideProject/comments/1jzoifx/eventure_ai_an_app_for_discovering_local_events/)

<sup>5</sup> <sup>6</sup> <sup>7</sup> <sup>8</sup> <sup>9</sup> <sup>10</sup> <sup>11</sup> <sup>12</sup> <sup>18</sup> <sup>19</sup> 10 Alternatives to Facebook for Local Event Discovery - Laparizone BLOG

<https://blog.laparizone.com/how-to-find-local-events-without-facebook/>

<sup>13</sup> Web Browser Tool | Langchain

<https://js.langchain.com/docs/integrations/tools/webbrowser/>

<sup>14</sup> Local Events API - Ranked Data Feed - PredictHQ

<https://www.predicthq.com/events/local-events>

<sup>15</sup> Integrating Web Search with Large Language Models using ...

[https://medium.com/@pankaj\\_pandey/integrating-web-search-with-large-language-models-using-langchain-a-complete-developer-guide-950bd75dcc2d](https://medium.com/@pankaj_pandey/integrating-web-search-with-large-language-models-using-langchain-a-complete-developer-guide-950bd75dcc2d)

<sup>16</sup> <sup>17</sup> Moving Statistics: What US Cities are People Moving to? {2025}

<https://raleighrealty.com/blog/moving-trends>