

# TixStarter

[Heroku link](#) **NB:** This should be a link to your production site

## Minimum Viable Product

TixStarter is a web application for creating events and buying tickets. Inspired by Kickstarter, events must achieve a revenue goal in order to be funded. The aesthetic design is inspired in part by TicketLeap, and in part by a desire to create a more beautiful ticketing app.

By the end of Week 9, this app will, at a minimum, satisfy the following criteria:

- New account creation, login, and guest/demo login
- Smooth, bug-free navigation
- Adequate seed data to demonstrate the site's features
- The minimally necessary features for an Evernote-inspired site: note creation and saving, note editing, and notes organized into notebooks
- Hosting on Heroku
- CSS styling that is satisfactorily visually appealing
- A production README, replacing this README

## Product Goals and Priorities

TixStarter will allow users to do the following:

- Create an account (MVP)
- Log in / Log out, including as a Guest/Demo User (MVP)
- Create, read, edit, and delete events (MVP)
- Create, read, edit, and delete showtimes for those events (MVP)
- Create, read, edit, and delete tickets for those showtimes (MVP)
- Simulate buying tickets (MVP)

## Design Docs

# Implementation Timeline

## Phase 1: Backend Set-Up, User Model, and Auth (0.5 Days)

**Objective:** Functioning rails project with Authentication

- create new project
- create User model
- authentication
- user signup/signin pages
- blank landing page after signin

## Phase 2: Events Model, API, and basic APIUtil (0.5 Days)

**Objective:** Events can be created, read, edited and destroyed through the API.

- create Event model
- seed the database with a small amount of test data
- CRUD API for events (EventController)
- jBuilder views for events
- setup Webpack & Flux scaffold
- setup APIUtil to interact with the API
- test out API interaction in the console.

## Phase 3: Flux Architecture and Router (0.5 Days)

**Objective:** Events can be created, read, edited and destroyed with the user interface.

- Setup the flux loop with skeleton files
- Setup React Router
- Implement each event component, building out the flux loop as needed.
  - EventsIndex
  - EventIndexItem
  - EventForm
  - EventDetail

## Phase 4: Showtimes (1 Day)

**Objective:** Events have many Showtimes.

- create Showtime model
- build out API, Flux loop, and components for:
  - Showtime CRUD
  - Fetching tickets
    - ShowtimesIndex
    - ShowtimeIndexItem
    - ShowtimeForm
    - ShowtimeDetail

## Phase 5: Tickets (1 Day)

**Objective:** Showtimes have many Tickets.

- create Tickets model and join table
- build out API, Flux loop, and components for:
  - Tickets CRUD
  - Fetching tickets
    - TicketsIndex
    - TicketIndexItem
    - TicketForm

## Phase 6: Aesthetics (4 days)

**Objective:** Make it sexy!

- Implement modals
- Style menus
- Style splashes
- Style buttons
- Style modals
- Ensure appropriate sizing for components
- Implement video embedding

Get feedback on my UI from others

## Phase 7: Styling Cleanup and Seeding (1 day)

**Objective:** Make the site feel clean and purposeful.

- Get feedback on my UI from others
- Refactor HTML classes & CSS rules

## Bonus Features (TBD)

- Instant Search for Events Index
  - Pagination / infinite scroll for Events Index
  - Animations on load
- Video background on homepage  
Animations for demo account login  
Parallax Scrolling