

Creating an Educational Chatbot with Replit AI

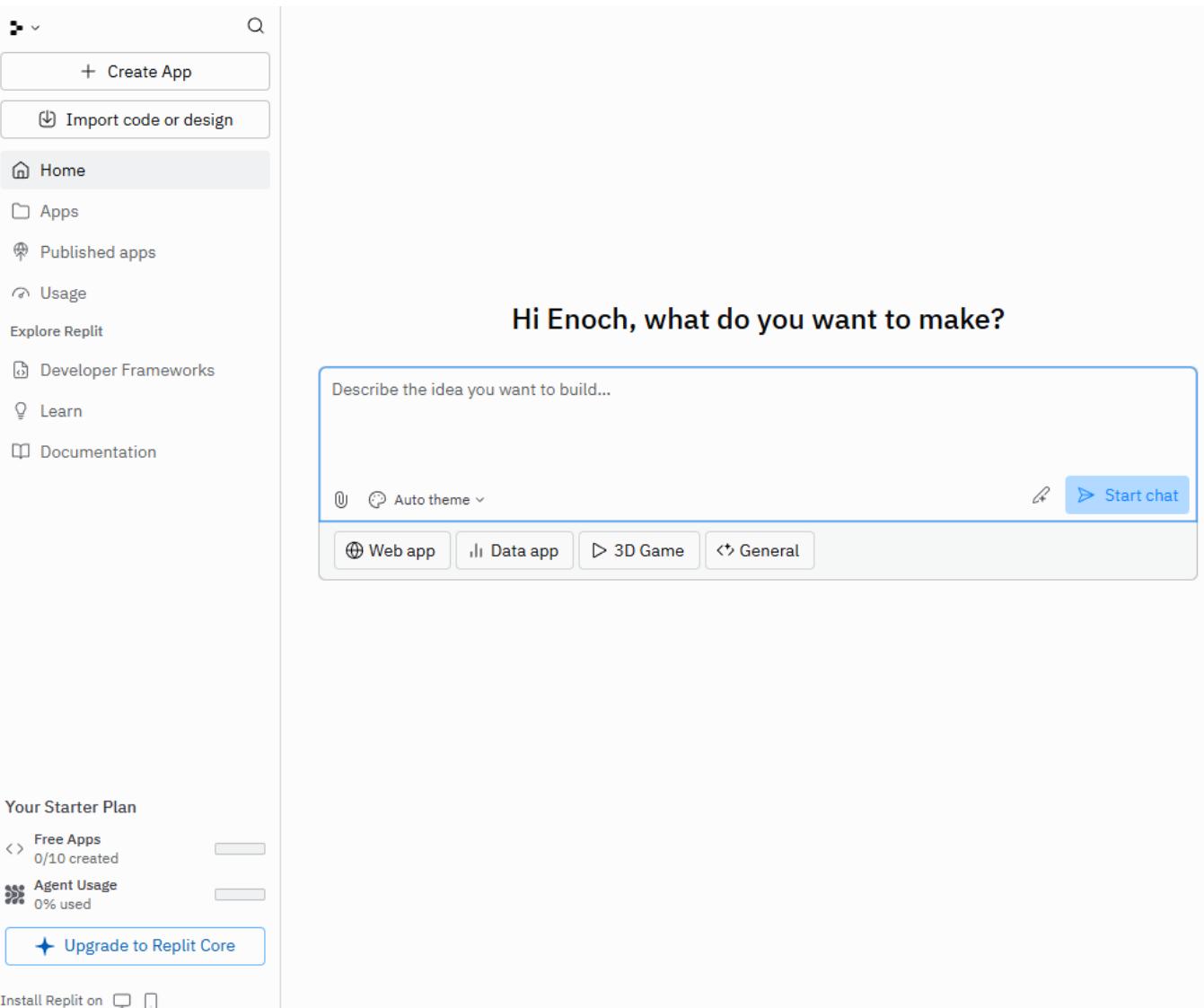
This tutorial demonstrates the step-by-step process of using Replit's AI features to create a fully functional educational chatbot application. The following screenshots capture the entire development workflow from initial prompt to deployment.

Tutorial Overview

This guide shows how to leverage Replit's AI-powered development environment to rapidly prototype and deploy an educational chatbot. The process involves prompt engineering, AI-assisted coding, debugging, and deployment.

Step-by-Step Development Process

Step 1: Getting Started



The screenshot shows the Replit AI interface. On the left, there is a sidebar with navigation links: '+ Create App', 'Import code or design', 'Home' (which is selected and highlighted in grey), 'Apps', 'Published apps', 'Usage', 'Explore Replit', 'Developer Frameworks', 'Learn', and 'Documentation'. Below these is a section titled 'Your Starter Plan' with metrics for 'Free Apps' (0/10 created) and 'Agent Usage' (0% used). A blue button at the bottom of this section says 'Upgrade to Replit Core'. At the very bottom of the sidebar, it says 'Install Replit on' followed by icons for Mac and Windows. The main area has a search bar at the top. Below it, a large text input field asks 'Describe the idea you want to build...'. To the right of this input field is a 'Start chat' button with a pencil icon. Below the input field are four buttons: 'Web app', 'Data app', '3D Game', and 'General'. The 'General' button is highlighted in blue. At the top center of the main area, the text 'Hi Enoch, what do you want to make?' is displayed.

Initial Setup

- Opening Replit's development environment

- Accessing the AI assistant feature
 - Preparing to create a new chatbot project
-

Step 2: Crafting the AI Prompt

The screenshot shows the Replit AI interface. On the left, there's a sidebar with various options like Home, Apps, Publish, Integrations, Usage, Explore, Develop, Learn, and Document. A message bar at the top says "Pasted-You-are-an-expert-AI-prompt-engineer-and-chatbot-developer-Your-task-is-to-create-a-fully-functional-1757579244126_1757579244126.txt". The main area contains the following text:

You are an expert AI prompt engineer and chatbot developer. Your task is to create a fully functional educational chatbot script using Python, leveraging the OpenRouter API for natural language processing and generation. The user has provided their OpenRouter API key: [insert your OpenRouter API key here, but replace this placeholder with the actual key when running the script].

The chatbot should focus on helping users improve their vocabulary and grammar skills through interactive conversations. Key features:

1. **Analysis of User Inputs**: For every user message or prompt, analyze the text to dynamically generate:
 - 3 new vocabulary words (uncommon or advanced words related to the user's input theme, with definitions, example sentences, and synonyms).
 - 3 idioms (relevant idioms that could enhance the user's expression, with explanations and examples).
 - 3 grammar concepts (identify potential grammar issues in the user's input or suggest improvements, such as tense usage, sentence structure, or punctuation rules, with explanations and corrected examples).Use the OpenRouter API to call a suitable language model (e.g., via the 'openai/gpt-4o' endpoint or similar) for this analysis. Make the analysis contextual to the user's message to make it educational and personalized.
2. **Interactive Learning Mode**: The chatbot should engage in a conversation where it responds to the user's query, then presents the 3 vocab words, 3 idioms, and 3 grammar concepts derived from the analysis. It should quiz the user on them optionally (e.g., ask for definitions or usages) to reinforce learning.
3. **Review Space**: Maintain a persistent review section or "notebook" in memory (using a dictionary or list in the script) that stores all previously generated vocab, idioms, and grammar concepts across the session. Allow users to command "review" or "show notebook" to display a summarized list of all items learned so far, grouped by category. Include options to quiz on past items or export the notebook as a text file.
4. **General Behavior**:

At the bottom, there's a status bar with "English.", "Runs on its own for up to 200 minutes," and a "Run" button.

Prompt Engineering

- Pasting the carefully crafted prompt for the educational chatbot
 - Defining the requirements and specifications
 - Setting up the AI assistant with clear instructions for what we want to build
-

Step 3: AI Planning Phase

Pasted-You-are...

Just now

I'll help you create an educational chatbot that analyzes user inputs and teaches vocabulary, idioms, and grammar. Let me search for the OpenRouter integration to handle the API connection.

Q Searched Replit's integrations for "OpenRouter"

Let me also search for OpenAI integrations since OpenRouter uses OpenAI-compatible endpoints:

Q Searched Replit's integrations for "OpenAI"

Based on your request for an educational chatbot using Python and OpenRouter API, I'll create a plan for this project. Since OpenRouter uses OpenAI-compatible endpoints, I can use the Python OpenAI integration and adapt it for OpenRouter.

Working...

Stop

Planning

AI Analysis and Planning

- The AI assistant analyzing the requirements
- Breaking down the project into manageable components
- Planning the architecture and approach for the educational chatbot

Step 4: Building the Application

Searched Replit's integrations for "OpenAI"

Based on your request for an educational chatbot using Python and OpenRouter API, I'll create a plan for this project. Since OpenRouter uses OpenAI-compatible endpoints, I can use the Python OpenAI integration and adapt it for OpenRouter.

Integrations  OpenAI

An educational Python chatbot that analyzes user inputs to teach vocabulary, idioms, and grammar through interactive conversations

I'll include the following features:

- Console-based chatbot interface with continuous conversation loop
- OpenRouter API integration for natural language processing
- Dynamic analysis of user inputs to generate vocabulary, idioms, and grammar concepts
- Interactive learning mode with contextual educational content
- Persistent review notebook storing all learned items across session
- Quiz functionality for reinforcing vocabulary and grammar learning
- Review command to display summarized learning history
- Notebook export functionality to save learning progress

Change plan

Get notified when your app is ready 

I've created a feature list based on your request. If everything looks good, we can start creating.

How do you want to continue?



Build the entire app

20+ mins

Best if you want Agent to build out the full functionality of your app



Start with a design

5-10 mins

Best if you want to see a design prototype first, then iterate on visuals or features

Start building ↑

Comprehensive Development

- AI generating the complete application structure
- Creating both frontend and backend components
- Implementing the core chatbot functionality

Step 5: Application Link Generation

Searched Replit's integrations for "OpenAI"

Based on your request for an educational chatbot using Python and OpenRouter API, I'll create a plan for this project. Since OpenRouter uses OpenAI-compatible endpoints, I can use the Python OpenAI integration and adapt it for OpenRouter.

Integrations ⓘ OpenAI

An educational Python chatbot that analyzes user inputs to teach vocabulary, idioms, and grammar through interactive conversations

I'll include the following features:

- Console-based chatbot interface with continuous conversation loop
- OpenRouter API integration for natural language processing
- Dynamically generate vocabulary, idioms, and grammar concepts
- Integrate contextual educational content
- Personalize learning all learned items across session
- Queue vocabulary and grammar learning
- Review summarized learning history
- Notify users to save learning progress



Change plan

Get notified when your app is ready X

I've created a feature list based on your request. If everything looks good, we can start creating.

How do you want to continue?



Build the entire app

20+ mins

Best if you want Agent to build out the full functionality of your app



Start with a design

5-10 mins

Best if you want to see a design prototype first, then iterate on visuals or features

Start building ↑

Deployment URL

- Replit generating a live application URL
- Making the chatbot accessible via web browser
- Setting up the hosting environment

Step 6: Build Process

The screenshot shows the StoryGenius application interface. At the top, there's a navigation bar with icons for file operations, a search bar, and a 'Publish' button. Below the navigation is a task list titled 'User information manually set' under 'Agent 3'. The first task, 'Prepare computer for new program', is expanded, showing a description about using OpenRouter API with Python OpenAI integration. It includes a '0 / 6' status indicator and a circular refresh icon. Below this, there's a section for 'Integrations' with 'OpenAI' selected, describing it as an educational Python chatbot. A list of features is provided, including a console-based interface, natural language processing, and various learning modes. A checkbox for 'Get notified when your app is ready' is present. In the bottom right corner of the main area, there's a purple hexagonal logo and the text 'Preview will be available soon'. Below this, a checkbox for 'Use Plan Mode to chat without code changes' is shown. On the left side, there's a 'Created task list' section with a single item: 'Adding support for python-3.11.' followed by a 'Stop' button. At the bottom, there's a terminal-like interface with a blue border, labeled 'Make, test, iterate...', showing a 'Build' button and other command-line options.

Application Assembly

- Replit compiling and building the application
- Installing dependencies and setting up the runtime environment
- Preparing the app for testing

Step 7: Debugging Session

The screenshot shows the Replit platform interface. At the top, there are two tabs: "StoryGenius - Replit" and "Educational Chatbot: Vocabulary". The URL in the address bar is "replit.com/@greentealatte64/StoryGenius". The main workspace on the left contains a workflow titled "Set up process and test chatbot" which is 5/6 complete. A message from "User information manually set Agent 3" says "Let me check the LSP diagnostics and fix any issues:". Below this, under "Checked code issues", it says "I need to fix the LSP errors first. The issue is with the variable `i` in the quiz loop - it might not be bound if the loop doesn't execute. Let me fix this:". There are two entries in the "Edited educational_chatbot.py" dropdown: "Edited educational_chatbot.py" and "Edited educational_chatbot.py". A message from "User information manually set Agent 3" says "Let me check if there are still LSP errors and then set up the workflow:". Below this, under "Checked code issues", it says "Great! The LSP errors are fixed. Now let me set up a workflow for running the chatbot and then test it:". A message from "Configured Educational Chatbot to run python educati..." says "Let me check the logs to see what error occurred:". Below this, under "Checked system logs", it says "I need to install the `requests` library first. Let me install it:". A message from "Installed requests" says "Good! Now let me restart the workflow to test if the chatbot runs:". Below this, under "Restarted Educational Chatbot", it says "Working..". At the bottom of the workspace, there is a text input field with placeholder "Make, test, iterate..." and a "Build" button. On the right side of the interface, there is a "Console" tab showing a workflow named "python educ...". The console output includes a welcome message: "Welcome to your Educational Chatbot! I'll help you improve your vocabulary and grammar through our conversations. After each message, I'll provide personalized learning content." It also lists commands: "review" or "show notebook": View all learned items; "quiz": Take a quiz on learned content; "export": Save your learning notebook to a file; "exit": End the session. A note at the bottom of the console says "Please insert your OpenRouter API key in the script before running! Edit the OPENROUTER_API_KEY variable at the top of the script."

Troubleshooting Issues

- Identifying and resolving runtime errors
- AI-assisted debugging process
- Fixing configuration and code issues

Step 8: API Key Configuration

The screenshot shows the Replit IDE interface with two tabs open: "StoryGenius - Replit" and "Educational Chatbot: Vocabulary".

Left Panel (StoryGenius - Replit):

- User information manually set Agent 3
- In progress tasks: 6 / 6
- Edited `educational_chatbot.py`
- Message: Let me check the current LSP diagnostics and then test the updated chatbot:
- Code status: Checked code issues
- Message: Great! No LSP errors. Now let me ask the user for their OpenRouter API key and set up the environment variable:
- Message: To use the educational chatbot, I need your OpenRouter API key. OpenRouter is a platform that provides access to various AI models through a unified API. You can get your API key by:

 1. Going to <https://openrouter.ai/>
 2. Creating an account or signing in
 3. Navigating to your API keys section
 4. Generating a new API key

- Message: The API key will be securely stored as an environment variable and used to analyze your messages and generate personalized vocabulary, idioms, and grammar lessons.

Right Panel (Educational Chatbot: Vocabulary):

- Workflows: python educ...
- Preview: Welcome to your Educational Chatbot! I'll help you improve your vocabulary and grammar through our conversations. After each message, I'll provide personalized learning content.
- Commands:
 - 'review' or 'show notebook': View all learned items
 - 'quiz': Take a quiz on learned content
 - 'export': Save your learning notebook to a file
 - 'exit': End the session
- Console: Please insert your OpenRouter API key in the script before running! Edit the `OPENROUTER_API_KEY` variable at the top of the script.

Bottom Left (Secrets):

Key	Value
OPENROUTER_API_KEY	[redacted]

Bottom Right:

- Build
- Run
- Up

Authentication Setup

- Adding the necessary API keys for the chatbot service
- Configuring environment variables
- Setting up secure API authentication

Step 9: Checkpoint Verification

The screenshot shows the StoryGenius platform interface. At the top, there's a navigation bar with 'StoryGenius' logo, a progress bar, and an 'Upgrade' button. Below it, a sidebar shows 'Completed tasks' (9/9) and a section titled 'How to Use' which says the chatbot is running and ready for interaction. It lists four steps: 1. Respond to your conversation naturally, 2. Analyze your input for educational opportunities, 3. Present vocabulary, idioms, and grammar lessons, 4. Save everything to your persistent learning notebook. A 'Available Commands' section lists: review or show notebook, quiz, export, and exit. The main area shows a 'Workflows' tab with a workflow named 'python educ...'. The workflow preview includes a welcome message: 'Welcome to your Educational Chatbot! I'll help you improve your vocabulary and grammar through our conversations. After each message, I'll provide personalized learning content.' It also lists commands: review or show notebook (View all learned items), quiz (Take a quiz on learned content), export (Save your learning notebook to a file), and exit (End the session). Below the workflow preview is a text input field labeled 'You:' with a placeholder 'Type a message to start a conversation'. On the left, there's a 'Checkpoints' section with a green checkmark indicating a checkpoint made 2 minutes ago, and a note to add detailed project documentation. There are 'Rollback here' and 'Changes' buttons. Below that is a 'Time spent' section showing '1 minute of work'. At the bottom, there's a 'Build' button and some other UI elements.

Progress Verification

- Testing the current state of the application
- Verifying that core functionality is working
- Ensuring the chatbot responds correctly

Step 10: Backend Application

StoryGenius  Upgrade

Set up the project configuration and dependencies... Agent 3

Completed tasks 9 / 9

2. Analyze your input for educational opportunities

3. Present vocabulary, idioms, and grammar lessons
4. Save everything to your persistent learning notebook

Available Commands

- [review or show notebook](#) - View all learned items
- [quiz](#) - Take a quiz on learned content
- [export](#) - Save notebook to a file
- [exit](#) - End the session

The chatbot automatically loads your previous learning progress when you restart it, so your educational journey continues seamlessly across sessions. Start chatting with it now to begin learning!

 Checkpoint made 5 minutes ago

Add detailed project documentation for the educational chatbot 

[Rollback here](#) [Changes](#)

Show less	^
Time worked	1 minute
Work done	4 actions
Agent Usage ^	
Reviewing Agent's code	\$0.06
General work	\$0.23

Make, test, iterate...

 Build 

Workflows  Show Only Latest  Clear Past Runs

python educ... Ask Agent... 

VOCABULARY WORDS:

1. **INQUIRE**
Definition: to seek information by asking a question
Example: I decided to inquire about his well-being.
Synonyms: ask, question
2. **SALUTATION**
Definition: a gesture or utterance made as a greeting or acknowledgment of another's arrival or departure
Example: The letter began with a warm salutation.
Synonyms: greeting, welcome
3. **AMICABLE**
Definition: characterized by friendliness and absence of discord
Example: Their conversation remained amicable despite their past disagreements.
Synonyms: friendly, cordial

IDIOMS:

1. "hit it off"
Meaning: to get along well with someone from the start
Example: They met at the party and immediately hit it off.
2. "break the ice"
Meaning: to do or say something to relieve tension or get conversation going in a social setting
Example: She told a joke to break the ice at the meeting.
3. "get off on the right foot"
Meaning: to start a relationship or activity positively
Example: I want to get off on the right foot with my new boss.

GRAMMAR CONCEPTS:

1. Question formation
Explanation: In English, questions are typically formed by inverting the subject and auxiliary verb.
Correct: How are you?
Incorrect: You are how?
2. Punctuation in questions

Server-Side Implementation

- Examining the backend architecture
- API endpoints and server configuration
- Database and service integrations

Step 11: Frontend Development

The screenshot shows the StoryGenius application interface. On the left, there's a sidebar with 'Completed tasks' (9/9) and a list of available commands: review or show notebook, quiz, export, and exit. Below that, a message from the chatbot discusses learning progress and provides a checkpoint note. On the right, the main workspace displays a 'Workflows' section for 'python educ...'. It includes sections for 'VOCABULARY WORDS', 'IDIOMS', and 'GRAMMAR CONCEPTS', each listing numbered items with definitions and examples. A message from the user asks about building a front-end app and using an OpenRouter model.

Completed tasks

3. Present vocabulary, idioms, and grammar lessons
4. Save everything to your persistent learning notebook

Available Commands

- review or show notebook - View all learned items
- quiz - Take a quiz on learned content
- export - Save notebook to a file
- exit - End the session

The chatbot automatically loads your previous learning progress when you restart it, so your educational journey continues seamlessly across sessions. Start chatting with it now to begin learning!

Checkpoint made 7 minutes ago

Add detailed project documentation for the educational chatbot

Show less

Time worked	1 minute
Work done	4 actions
Agent Usage	\$0.29
Reviewing Agent's code	\$0.06
General work	\$0.23

can you build the front end app and also tell me which openrouter model are you using?

User Interface Creation

1. INQUIRE
Definition: to seek information by asking a question
Example: I decided to inquire about his well-being.
Synonyms: ask, question

2. SALUTATION
Definition: a gesture or utterance made as a greeting or acknowledgment of another's arrival or departure
Example: The letter began with a warm salutation.
Synonyms: greeting, welcome

3. AMICABLE
Definition: characterized by friendliness and absence of discord
Example: Their conversation remained amicable despite their past disagreements.
Synonyms: friendly, cordial

IDIOMS:

1. "hit it off"
Meaning: to get along well with someone from the start
Example: They met at the party and immediately hit it off.
2. "break the ice"
Meaning: to do or say something to relieve tension or get conversation going in a social setting
Example: She told a joke to break the ice at the meeting.
3. "get off on the right foot"
Meaning: to start a relationship or activity positively
Example: I want to get off on the right foot with my new boss.

GRAMMAR CONCEPTS:

1. Question formation
Explanation: In English, questions are typically formed by inverting the subject and auxiliary verb.
Correct: How are you?
Incorrect: You are how?
2. Punctuation in questions

Step 12: Complete Application

- Building the chatbot interface
- Implementing the chat UI components
- Creating an intuitive user experience

The screenshot shows two side-by-side interfaces. On the left is the StoryGenius project configuration interface, which includes a sidebar with 'Completed tasks' (6 / 6) and a main area listing files like app.py, templates/index.html, static/css/style.css, static/js/app.js, and learning_notebook.json. It also shows a 'Project Structure' section and a message about saving progress. On the right is the Replit interface for the 'Educational Chatbot' application. The browser tab shows '.replit.dev/' with a 'Ready to share with the world?' button and a 'Publish' button. The application itself has a blue header with the title 'Educational Chatbot' and a sub-header 'Learn vocabulary, idioms, and grammar while you chat!'. A notification bubble indicates '9 items learned'. The main content area features a bot icon and a welcome message: 'Hello! I'm your Educational Chatbot. Send me a message and I'll help you learn new vocabulary, idioms, and grammar concepts!'. Below this is a 'Tips:' section with three bullet points: 'Write detailed messages (at least 3 words) for better learning content', 'Check your notebook to review what you've learned', and 'Take quizzes to test your knowledge'. At the bottom is a message input field with placeholder 'Type your message here...' and a send button, along with '0/500 characters' and buttons for 'Notebook' and 'Quiz'.

Integrated System

- The complete educational chatbot application
- Frontend and backend working together
- Full chat functionality implemented

Step 13: Application Testing

The image shows two side-by-side screenshots of software interfaces. On the left is the StoryGenius platform, which includes a navigation bar with 'StoryGenius' and 'Upgrade' buttons, a sidebar with 'Completed tasks' (6 / 6), and a main content area with bullet points about learning progress and a detailed 'Project Structure' section. Below this is a section about automatic saves and API storage, followed by a message encouraging users to start chatting. At the bottom is a 'Checkpoint made 2 minutes ago' section with a note to add an introductory message, a screenshot of a chatbot interface, and buttons for 'Rollback here' and 'Changes'. There's also a '6 minutes of work' timer and a 'Make, test, iterate...' input field with a 'Build' button.

The right screenshot shows the Replit 'Preview' tab with a URL like '.replit.dev/'. It displays a 'Your Learning Content' page with a 'Vocabulary Words' section. The first word is '1. COMMUTE', defined as 'to travel some distance between one's home and place of work on a regular basis', with an example sentence and synonyms. The second word is '2. TRANSPORTATION', defined as 'the action of transporting someone or something or the process of being transported', with an example sentence and synonyms. The third word is '3. INFRASTRUCTURE', defined as 'the basic physical and organizational structures and facilities needed for the operation of a society or enterprise', with an example sentence.

Quality Assurance

- Comprehensive testing of the chatbot features
- Verifying educational content delivery
- Testing user interactions and responses

Step 14: Streaming Bug Discovery

The screenshot shows two main windows. On the left is the StoryGenius application interface, which includes a header with 'StoryGenius' and an 'Upgrade' button. Below the header, there's a list of tasks under 'Agent 3': 'Set up the project configuration and dep...' and 'Learning notebook system is operational'. A section titled 'API Response:' shows the EdUHK API responding with a non-streaming mode message. Another section discusses Mock Mode vs Proxy Mode. On the right is a browser window displaying a preview of a web application titled 'Educational Chatbot'. The page says 'Ready to share with the world?' and has a 'Publish' button. It features a graduation cap icon and a message: 'Learn vocabulary, idioms, and grammar while you chat!'. A progress bar indicates '27 items learned'. A message box contains the text 'The more you write, the better I can help you learn new vocabulary and grammar!'. Below this, a user message says 'I would like to express my gratitude to my beloved parents' and a bot response says 'Response from EdUHK API (non-streaming mode)'. At the bottom, there's a text input field with placeholder 'Type your message here...' and a send button with a paper airplane icon. The input field shows '0/500 characters'. The bottom navigation bar includes tabs for 'Notebook' and 'Quiz', and a 'Learning Notebook' section with a 'VOCABULARY WORDS' card.

Issue Identification

- Discovering problems with real-time message streaming
- Identifying performance bottlenecks
- Analyzing streaming response issues

Step 15: Bug Resolution

StoryGenius

Checkpoint made

Completed tasks 4 / 4

Full Functionality Restored:

- Chat responses are natural and educational
- Web interface works perfectly
- Learning notebook tracks progress
- Quiz and review systems operational

Ready for Use

Your educational chatbot is now:

- Fully compatible with EdUHK's streaming-only API
- Generating real content instead of mock responses
- Web application running smoothly at port 5000
- All features working - chat, learning, quizzes, and notebook

The streaming implementation successfully connects to your EdUHK API and delivers the full educational experience you designed! 🎉

Checkpoint made 1 minute ago

Improve chatbot's response generation with varied output formats

Rollback here Changes

3 minutes of work

Checkpoint made just now

Add new vocabulary, idioms, and grammar concepts to the learning notebook

Rollback here Changes

sorry I

Preview x >_ Console x + Q & Publish

Ready to share with the world? Publish

Your Learning Content

Vocabulary Words

1. GRATITUDE

Definition: the quality of being thankful; readiness to show appreciation for and to return kindness.

Example: *Her gratitude was evident in the way she thanked everyone who helped her.*

Synonyms: thankfulness, appreciation

2. BELOVED

Definition: dearly loved.

Example: *She wrote a heartfelt letter to her beloved mentor.*

Synonyms: cherished, dear

3. EXPRESS

Definition: to convey (a thought or feeling) in words or by gestures and conduct.

Example: *He found it difficult to express his feelings in front of others.*

Synonyms: articulate, communicate

Problem Solving

- Implementing fixes for the streaming issues
 - Optimizing real-time communication
 - Ensuring smooth message delivery

Step 16: Token Management

The screenshot shows the Replit homepage. On the left, there's a sidebar with links like Home, Apps, Published apps, Integrations, Usage, Explore Replit, Developer Frameworks, Learn, and Documentation. Below that is a section for 'Your Starter Plan' showing 'Free Apps' (1/10 created) and 'Agent Usage' (49% used), with a 'Upgrade to Replit Core' button. At the bottom of the sidebar are links to install Replit on desktop and mobile. The main area has a search bar at the top and a large central box asking 'Hi Enoch, what do you want to make?'. It includes a text input field 'Describe the idea you want to build...', a theme switcher, and a 'Start chat' button. Below this are buttons for Web app, Data app, 3D Game, General, Agents & Automations (which is highlighted in blue), and Documentation. A prominent banner at the bottom left says 'New Introducing Agent 3: Autonomy for all' with a video thumbnail for 'Introducing Agent 3'.

Authentication Tokens

- Managing API tokens and authentication
- Configuring secure access credentials
- Setting up token-based authorization

Step 17: Publication Process

The screenshot shows two adjacent application windows. On the left is the StoryGenius interface, which includes a sidebar with 'Completed tasks' and a main area displaying a checklist of successful features. A 'Ready for Use' section indicates the chatbot is functional. Below this are two card-like sections for checkpoints, each with a 'Rollback here' button and a 'Changes' link. At the bottom of the StoryGenius window is a toolbar with icons for 'Build', 'Deploy', and 'Edit'. On the right is the Replit 'Publish' interface, featuring a 'Primary URL' input field containing 'story-genius-greentealatte64.replit.app' with a green checkmark indicating it's available. A 'Publish' button is visible at the top right of the Replit window.

StoryGenius

Completed tasks

Chat responses are natural and educational

- Web interface works perfectly
- Learning notebook tracks progress
- Quiz and review systems operational

Ready for Use

Your educational chatbot is now:

- Fully compatible with EdUHK's streaming-only API
- Generating real content instead of mock responses
- Web application running smoothly at port 5000
- All features working - chat, learning, quizzes, and notebook

The streaming implementation successfully connects to your EdUHK API and delivers the full educational experience you designed! 🎉

Checkpoint made 10 minutes ago

Improve chatbot's response generation with varied output formats

Rollback here Changes

3 minutes of work

Checkpoint made 8 minutes ago

Add new vocabulary, idioms, and grammar concepts to the learning notebook

Rollback here Changes

Make, test, iterate...

Build Deploy Edit

Preview Public... Console + Q Publish

Publish your app

Primary URL

You can add your own custom domain after publishing your app.

story-genius-greentealatte64.replit.app

✓ Available

Upgrade your plan to publish

What does publishing do?

⊕ Publishing your app makes it available for anyone to use
Your friends and users around the world can visit your app through your primary URL.

\$ Costs are included in Core subscriptions
Costs vary depending on the type of technology used to publish your app.

Watch video Learn more

Deployment Preparation

- Preparing the application for public deployment
- Configuring production settings
- Publishing to the live environment

Step 18: Educational Chatbot Ready

The screenshot shows the StoryGenius application interface. On the left, there's a sidebar with a tree view showing 'Completed tasks' (4 / 4) and a list of bullet points: 'Chat responses are natural and educational', 'Web interface works perfectly', 'Learning notebook tracks progress', and 'Quiz and review systems operational'. Below this is a section titled 'Ready for Use' with the subtext 'Your educational chatbot is now:' followed by a bulleted list: 'Fully compatible with EdUHK's streaming-only API', 'Generating real content instead of mock responses', 'Web application running smoothly at port 5000', and 'All features working - chat, learning, quizzes, and notebook'. A note below states: 'The streaming implementation successfully connects to your EdUHK API and delivers the full educational experience you designed!' with a graduation cap emoji.

On the right, the main area is titled 'Publish your app'. It has a 'Primary URL' input field containing 'education-chatbot-2025.replit.app' with a green checkmark indicating it's available. There's a blue button to 'Upgrade your plan to publish'. Below this is a collapsed section titled 'What does publishing do?' which, when expanded, shows two items: 'Publishing your app makes it available for anyone to use' (with a note about friends and users around the world) and 'Costs are included in Core subscriptions' (with a note about varying costs based on technology). Buttons for 'Watch video' and 'Learn more' are also present.

Final Product

- The completed educational chatbot application
- Ready for student and educator use
- Fully functional and deployed

Step 19: Adding Secrets

The screenshot shows the Replit account dashboard. On the left, there's a sidebar with links like Account, Profile, Notifications, Create Team, CLUI, Theme, Help, Log out, Developer Frameworks, Learn, and Documentation. Below this is the 'Your Starter Plan' section, which includes metrics for Free Apps (1/10 created) and Agent Usage (49% used), and a 'Upgrade to Replit Core' button. A URL <https://replit.com/account> is visible at the bottom of the sidebar.

The main area has a search bar and a message 'Hi Enoch, what do you want to make?'. It features a text input field 'Describe the idea you want to build...', a theme switcher, and a 'Start chat' button. Below these are buttons for Web app, Data app, 3D Game, General, Agents & Automations (Beta), and a 'Learn more' button.

A 'New' notification box is prominently displayed, titled 'Introducing Agent 3: Autonomy for all'. It lists three features: App Testing, Agents & Automations, and Agent 3 autonomy. To the right of the text is a video thumbnail for 'Introducing Agent 3'.

Security Configuration

- Adding sensitive configuration as environment secrets
- Securing API keys and credentials
- Implementing proper security practices

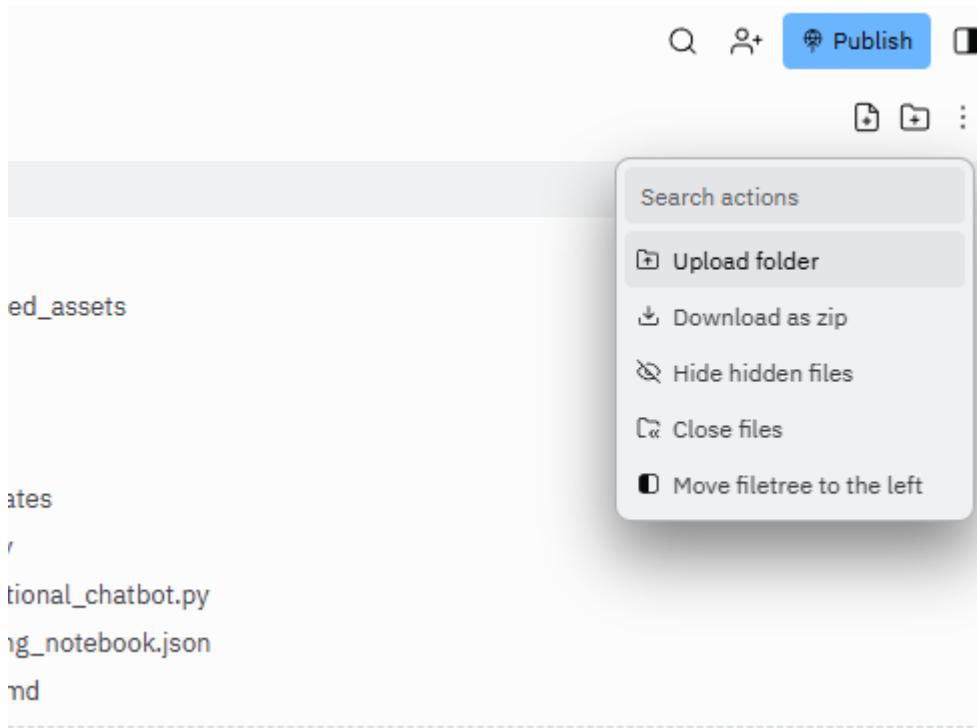
Step 20: Download Process - Part 1



Export Preparation

- Beginning the process to download the project
 - Preparing the codebase for local development
 - Setting up export options
-

Step 21: Download Process - Part 2



Project Export

- Completing the download of the entire project
 - Obtaining the source code for local development
 - Enabling offline development and customization
-

🔗 Key Takeaways

AI-Powered Development Benefits:

1. **Rapid Prototyping:** From concept to working application in minutes
2. **Automated Code Generation:** AI handles boilerplate and complex implementations
3. **Intelligent Debugging:** AI assists in identifying and fixing issues
4. **End-to-End Solution:** Complete application with frontend, backend, and deployment

Educational Chatbot Features:

- Interactive learning conversations
- Educational content delivery
- Real-time streaming responses
- User-friendly interface
- Secure API integration

Replit Platform Advantages:

- **Zero Setup:** No local development environment needed
- **Instant Deployment:** Live URLs generated automatically
- **Collaborative:** Easy sharing and collaboration
- **Integrated Tools:** Built-in debugging, testing, and deployment

Development Workflow:

1. **Prompt Engineering:** Clear, detailed requirements
2. **AI Generation:** Automated code and structure creation
3. **Iterative Testing:** Continuous verification and improvement
4. **Debugging:** AI-assisted problem resolution
5. **Deployment:** Seamless publishing to production
6. **Security:** Proper secret and credential management

Technical Implementation

Architecture Components:

- **Frontend:** Interactive chat interface
- **Backend:** API server with chatbot logic
- **Authentication:** Secure API key management
- **Streaming:** Real-time message delivery
- **Deployment:** Cloud-hosted application

Key Technologies:

- Replit's AI development assistant
- Real-time streaming protocols
- RESTful API design
- Environment variable management
- Token-based authentication

Educational Applications

Use Cases:

- **Student Support:** 24/7 educational assistance
- **Course Content:** Interactive learning materials
- **Assessment:** Automated quiz and feedback systems
- **Research:** Information discovery and explanation

Benefits for Education:

- **Accessibility:** Always available learning support
- **Personalization:** Adapted responses for individual needs
- **Scalability:** Serves multiple students simultaneously
- **Cost-Effective:** Reduces need for human tutoring

This tutorial demonstrates the power of AI-assisted development in creating practical educational tools. The entire process from conception to deployment can be completed in a matter of hours, making it an excellent approach for rapid educational technology development.